



minsait
by Indra

Desarrollo de aplicaciones utilizando arquitecturas de nueva generación

Autor

Miguel Salamero Muñoz

Ponente

Fatos Xhafa

Dept. de Ciències de la Computació

Directora

Mónica Herrera Jiménez

Indra Sistemas

Especialidad

Ingeniería del Software

Grado en Ingeniería Informática

2 de julio de 2020

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

Resumen

Este proyecto consiste en la mejora, aumento y progreso de la funcionalidad, navegación y ergonomía de las operativas de financiación de pagos y créditos documentarios para la aplicación CaixaBankNow de La Caixa. Este trabajo académico de fin de grado se ha desarrollado con la colaboración de la empresa Minsait con el empleo de metodologías ágiles.

En primera instancia se define el problema y se detallan los objetivos y requisitos. Posteriormente, se elabora un informe del alcance, de la planificación temporal y del presupuesto económico del proyecto. Finalmente, en las etapas de diseño, arquitectura e implementación, se desarrolla la aplicación, se argumentan las decisiones de diseño tomadas y se expone cómo ha ido evolucionando el proyecto en cada iteración.

Como resultado, se ha obtenido una nueva versión completa de la operativa de solicitud de anticipos de facturas que es resolutive, eficiente, ergonómica y útil para los usuarios. Esta modificación permitirá realizar las operaciones financieras de una forma más ágil y rápida. El proyecto conlleva un cambio en la estructura y diseño del software, logrando como resultado un código correcto, depurado y de fácil mantenimiento.

Resum

Aquest projecte consisteix en la millora, augment i progrés de la funcionalitat, navegació i ergonomia de les operatives de finançament de pagaments i crèdits documentaris per a l'aplicació CaixaBankNow de La Caixa. Aquest treball acadèmic de fi de grau s'ha desenvolupat amb la col·laboració de l'empresa Minsait amb l'ocupació de metodologies àgils.

En primera instància es defineix el problema i es detallen els objectius i requisits. Posteriorment, s'elabora un informe de l'abast, de la planificació temporal i del pressupost econòmic del projecte. Finalment, a les etapes de disseny, arquitectura i implementació, es desenvolupa l'aplicació, s'argumenta la pressa de decisions de disseny i s'exposa com ha anat evolucionant el projecte en cada iteració.

Com a resultat, s'ha obtingut una nova versió completa de l'operativa de sol·licitud de bestretes de factures que és resolutiva, eficient, ergonòmica i útil pels usuaris. Aquesta modificació permetrà realitzar les operacions financeres d'una forma més àgil i ràpida. El projecte comporta un canvi en l'estructura i disseny del programari, aconseguint com a resultat un codi correcte, depurat i de fàcil manteniment.

Abstract

This project consists of improving, increasing and progressing the functionality, navigation and ergonomics of the operations for financing payments and documentary credits for the CaixaBankNow application by La Caixa. This end-of-degree academic project has been carried out in collaboration with the company Minsait using agile methodologies.

In the first instance, the problem is defined and the objectives and requirements are detailed. Subsequently, a report on the scope, temporary planning and economic budget of the project is prepared. Finally, in the design, architecture and implementation stages, the application is developed, the design decisions made are discussed and it is exposed how the project has evolved in each iteration.

As a result, a new complete version of the invoice advance request operation has been obtained that is decisive, efficient, ergonomic and useful for users. This modification will allow financial operations to be carried out in a more agile and faster way. The project involves a change in the structure and design of the software, resulting in correct, clean and easy-to-maintain code.

Índice

1.	Contexto	1
1.1.	Introducción	1
1.2.	Definición de conceptos.....	1
1.3.	Problema a resolver	3
1.4.	Actores implicados	4
2.	Justificación.....	6
3.	Alcance.....	8
3.1.	Objetivos del proyecto	8
3.1.1.	Objetivos principales	8
3.1.2.	Objetivos secundarios	8
3.2.	Requisitos.....	9
3.2.1.	Requisitos funcionales	9
3.2.2.	Requisitos no funcionales	10
3.3.	Obstáculos y riesgos	11
4.	Metodología y rigor.....	12
4.1.	Herramientas de seguimiento.....	14
5.	Duración del proyecto	15
6.	Planificación temporal	16
6.1.	Definición y estimación de las tareas	16
6.1.1.	Gestión del proyecto:	16
6.1.2.	Sprints.....	17
6.1.3.	Seguimiento y control del proyecto	19
6.1.4.	Documentación del proyecto	20
6.2.	Resumen de tareas	21

6.3.	Diagrama de Gantt	22
7.	Gestión del riesgo y planes alternativos	23
8.	Presupuesto	25
8.1.	Identificación de costes	25
8.1.1.	Costes de personal por actividad	25
8.1.2.	Costes genéricos	27
8.1.3.	Contingencias	27
8.1.4.	Imprevistos	28
8.1.5.	Costes totales	28
8.2.	Control de gestión	29
9.	Sostenibilidad	31
9.1.	Autoevaluación	31
9.2.	Dimensión económica	32
9.3.	Dimensión ambiental	32
9.4.	Dimensión social	32
10.	Arquitectura y diseño	34
10.1.	Patrones arquitectónicos	34
	Modelo vista controlador	34
10.2.	Patrones de diseño	36
	Data transfer Object	36
	Business Delegate	36
	Inyección de dependencias	36
	Factoría	37
	Singleton	37
10.3.	Herramientas utilizadas	38
11.	Implementación	40
11.1.	Sprints	43

Sprint 1 y 2	43
Sprint 3	43
Sprint 4	47
Sprint 5	48
Sprint 6	51
Sprint 7	52
Sprint 8	54
Sprint 9	55
11.2. Testing.....	56
11.3. Integración y entrega continua	57
12. Conclusiones	59
12.1. Consolidación de las competencias técnicas.....	60
12.2. Trabajo futuro.....	62
13. Referencias	63
Anexo A. Diagrama de Gantt	66

Índice de tablas

Tabla 1. Resumen de tareas	21
Tabla 2. Coste por hora según tipo perfil	25
Tabla 3. Coste de personal por actividad.....	26
Tabla 4. Costes genéricos	27
Tabla 5. Costes totales	28
Tabla 6. Costes genéricos recalculados	30

Índice de figuras

Figura 1. Modelo Vista Controlador.....	34
Figura 2. Diagrama de secuencia - Petición	35
Figura 3. Patrón factoría	37
Figura 4. Patrón singleton.....	38
Figura 5. Confirming CaixaBankNow	41
Figura 6. Solicitud de cobro anticipado - Selección de cliente.....	41
Figura 7. Confirming Petición 19 - Versión antigua	45
Figura 8. Confirming Petición 19 - Versión actual.....	45
Figura 9. Confirming Petición 21 - Versión antigua 1	49
Figura 10 . Confirming Petición 21 - Versión antigua 2	50
Figura 11. Confirming Petición 21 - Versión actual.....	50
Figura 12. Confirming Petición 25 - Versión antigua	51
Figura 13 . Confirming Petición 25 - Versión actual.....	52
Figura 14. Confirming Petición 27 - Versión antigua	53
Figura 15. Confirming Petición 27 - Versión actual.....	54
Figura 16. Jenkins	57

1. Contexto

1.1. Introducción

Este trabajo de fin de grado titulado “Desarrollo de Aplicaciones utilizando arquitecturas de nueva generación” pertenece a la especialidad de Ingeniería de Software de los estudios del Grado de Ingeniería Informática de la Universitat Politècnica de Catalunya realizados en la Facultad de Informática de Barcelona [1].

El proyecto se realiza con la cooperación de la empresa INDRA Soluciones Tecnológicas de la Información [2] y trata de la mejora de la aplicación de banca online CaixaBankNow, de la entidad financiera La Caixa [3]. Este conjunto de mejoras se segmenta en el aumento y progreso de la funcionalidad, navegación y ergonomía de las operativas de financiación de pagos y créditos documentarios de la entidad bancaria. El resultado final del proyecto será una renovación por completo de la experiencia del usuario en la aplicación CaixaBankNow. Todas las modificaciones que se efectúan en este proyecto afectarán a la modalidad para empresas de la plataforma de banca online. Este trabajo entra dentro del gran paquete de mejoras y de rediseño que se ha anunciado para CaixaBankNow este 2020.

1.2. Definición de conceptos

1. Confirming

Confirming es un servicio de CaixaBankNow que consiste en la gestión de los pagos de una empresa a todos sus proveedores [4]. Se encarga de notificar a los proveedores las órdenes de pago, ofreciendo opcionalmente la posibilidad de anticiparlos sin trámites bancarios. Permite gestionar los pagos a proveedores con agilidad.

Es también el proyecto de negocio en donde se enmarca, principalmente, este trabajo.

2. FwkLO

FwkLO es el nuevo *framework* de *Frontend* introducido por la Caixa en 2019 para el desarrollo del proyecto. Tiene el objetivo de desacoplar las lógicas de cada operativa de negocio y de hacer más claro y fácil el flujo de datos entre proyectos. Está basado en el modelo vista controlador y se asemeja en distintos apartados al *framework Spring*.

3. Palanca

La palanca es un método que sirve para dirigir el flujo de las operativas de la aplicación. Son atributos que se declaran en ficheros de propiedades – uno por cada PN – y que pueden estar activos o no. Con la incorporación de nuevas funcionalidades se suelen crear nuevas palancas. El código correspondiente a la nueva operativa se encapsula dentro de una cláusula if-else que comprueba el valor de la palanca asociada a esa historia de usuario.

De esta manera se consigue que en cualquier momento se pueda restaurar la funcionalidad antigua o redirigir el flujo de pantallas en tiempo de ejecución. Simplemente es necesario modificar el fichero de propiedades y cambiar el valor de las palancas.

4. Petición. PE

Dentro de un proyecto de negocio se encuentran diversas peticiones que hacen referencia a funcionalidades concretas que puede realizar un usuario. A cada una de estas peticiones se le asigna un número. Por lo tanto, dentro de cada PN se localizan unos determinados PE. Siguiendo con el ejemplo anterior, una pantalla concreta – la petición 3 - de la operativa de Créditos Documentarios se le llamaría el PE 3 del PN CDO.

5. Proyecto de Negocio. PN

Proyecto de Negocio, normalmente identificado con tres siglas. Dentro de un PN se recoge un conjunto de operativas que tienen relación entre sí. Un ejemplo podría ser CDO, donde se reúnen aquellas funcionalidades que pertenecen a la operativa de Créditos Documentarios. Un proyecto de negocio equivale en el desarrollo a un proyecto Java.

6. Diseño de Experiencia de Usuario. UX Design

El diseño de experiencia del usuario es el proceso de diseño que se utiliza para crear productos que brindan experiencias significativas, relevantes y confortables para los usuarios. Esto implica el diseño de todo el proceso de integración del producto, incluidos los aspectos de marca, diseño, usabilidad y función.

1.3. Problema a resolver

La plataforma online CaixaBankNow ofrece a sus clientes la oportunidad de realizar cualquier trámite u operación bancaria de forma fácil e inmediata. Es utilizada a diario por millones de personas en todo el mundo y está disponible a cualquier hora y día del año. En una aplicación tan importante, ampliamente extendida y tan utilizada como lo es ésta, es deseable que ofrezca siempre un servicio cómodo, seguro, claro, facilitador e intuitivo para el usuario.

Es por este motivo que el principal problema a resolver del proyecto es la mejora de la experiencia del usuario en el uso habitual de la plataforma CaixaBankNow. En Indra Soluciones Tecnológicas de la Información desarrolla dichas mejoras en las operativas de financiación y de créditos documentarios.

CaixaBankNow, antes llamada Línea Oberta, es una plataforma que fue lanzada al mercado hace más de diez años. Parte de su software está basado en tecnologías que, con el paso del tiempo, han quedado en desuso o, directamente, obsoletas. Además de esto, existen distintas pantallas, sobre todo en la versión para empresas, que presentan un diseño envejecido, con una navegación confusa e imprecisa y que no concuerdan con el resto de la aplicación.

El pasado año 2019, La Caixa, anunció un nuevo conjunto de mejoras de diseño de su plataforma online que tenían como objetivo hacer que el usuario experimentara una sensación de mejora en la utilización de la aplicación. Juntamente con este anuncio, se introdujo un nuevo framework de desarrollo para su plataforma llamado *FwkLO*. Por lo tanto, en este documento se exponen todos los cambios ejecutados que han sido necesarios para mejorar las diferentes operativas de negocio, adoptando dichos cambios a la nueva arquitectura y framework.

En definitiva, el objetivo final del proyecto es modificar y actualizar distintas operativas de la banca online para empresas que mejoren la experiencia del usuario. Todo ello conlleva la mejora y perfeccionamiento en el diseño de las pantallas, cambios en la lógica de negocio y el traspaso del software antiguo al nuevo. Desde CaixaBank se requiere que toda la aplicación de CaixaBankNow tenga una interfaz significativa y que consiga mejorar la vida de los usuarios [5]. Por lo tanto, en este proyecto se debe compartir el mismo objetivo y conseguir que no solo las interfaces sean confortables y atractivas, sino que también ayuden y faciliten el trabajo a los usuarios.

1.4. Actores implicados

1. Directora y ponente

La directora del proyecto de fin de grado es Mónica Raquel Herrera. El ponente del trabajo, Fatos Xhafa, es miembro del departamento de ciencias de la computación y, aparte de estar involucrado en el correcto avance del trabajo, fue quién me puso en contacto con la empresa Indra.

2. Equipo de desarrollo

Equipo del departamento de tecnologías avanzadas de Minsait encargado del desarrollo y test de las historias de usuarios determinadas por SILK. El objetivo de este actor es llevar a cabo la implementación de las historias de usuario.

3. Minsait

Empresa de Indra que asume el proyecto de mejora de experiencia de usuario de la plataforma CaixaBankNow por parte de La Caixa. Es responsable del correcto funcionamiento y de la mejora de ciertos proyectos de negocio de la aplicación.

4. SILK

SILK es la filial tecnológica de La Caixa [6]. Es, por tanto, el cliente externo y es quien contrata los servicios tecnológicos de Indra para el desarrollo del proyecto. Así mismo, son quienes redactan las historias de usuario y criterios de aceptación y quienes valoran el producto final. Con la finalización de este proyecto, tendrán un nuevo rediseño de pantallas que poder ofrecer a los clientes de CaixaBank.

5. Usuario final

Son todos los clientes del banco La Caixa que utilicen la aplicación de banca digital CaixaBankNow. En concreto aquellos usuarios que accedan a la aplicación en la modalidad de empresa. Estos clientes se verán beneficiados de la renovación de la experiencia de usuario en la aplicación. Podrán realizar las operaciones bancarias de forma fácil y ágil obteniendo una mejor respuesta por parte de la plataforma. Además, disfrutarán de nuevas funcionalidades para disponer de un mayor control sobre sus finanzas.

2. Justificación

Como se ha explicado en la introducción del documento, este es un proyecto realizado con un convenio de cooperación con la empresa INDRA Soluciones Tecnológicas de la Información. El problema a resolver es concreto y viene marcado por el cliente externo La Caixa de forma detallada. Por lo tanto, el diseño y estructura de la aplicación y las herramientas utilizadas para este trabajo son aquellas que se usan dentro del contexto manifestado.

Sin embargo, sí que es posible valorar alternativas de diseño. Para este problema en concreto se utiliza un framework propio llamado *FwkLO*. Es un *framework* que se asemeja en ciertos aspectos al conocido *Spring*. Permite trabajar con inversión del control y con inyección de dependencias. Además, provee de un sistema para el traspaso de datos entre clases en un mismo flujo. Es un hecho diferencial muy interesante el uso de un *framework* propio ya que se adaptará específicamente para la aplicación que estamos desarrollando y tendrá en cuenta restricciones y factores propios. Un aspecto negativo es que, al ser un framework de elaboración propia, todos los desarrolladores deberán aprenderlo desde cero para llegar a dominarlo eficazmente. No obstante, esto no es un problema grave porque como ya se ha comentado es similar a *Spring*.

Otra opción a tener en cuenta, que se podría complementar e integrar al sistema actual, sería la separación clara y explícita entre *backend* y *frontend*. En esta arquitectura sí que se trabaja con el modelo vista controlador, pero aun así se juntan entre sí esas dos partes del software en gran medida. Es por ello que sería una gran idea que la parte de frontend fuera independiente pudiendo realizarse con un framework y lenguaje totalmente distinto. Las principales opciones para aplicar este cambio en la arquitectura serían *Angular* y *React*. Con estas dos tecnologías el propio navegador del cliente dispone de los documentos HTML y ejecuta el *JavaScript* necesario para la interacción con la web. Por el contrario, en este proyecto se usan *JavaServer Pages* (jsp) para crear las vistas.

JavaServer Pages es una tecnología que permite crear páginas web dinámicas usando el lenguaje de programación Java. Otra opción que podría utilizarse para integrar en el modelo vista controlador de la arquitectura y reemplazar a JSP sería Thymeleaf. Esta tecnología es una biblioteca de Java que actúa como motor de plantillas HTML5. Al

trabajar con HTML hace que el código sea mucho más legible que las jsp y no añade nuevas etiquetas, sino que añade nuevos atributos.

Sin embargo, la gran ventaja de JSP respecto Thymeleaf es su gran rendimiento. Esto se produce debido a que el código es compilado como cualquier otra clase Java. Esto hace que JSP sea más eficiente que otras tecnologías web que ejecutan el código de una manera puramente interpretada.

Pese a que puedan existir otras diferentes alternativas más actuales que ofrezcan un mejor rendimiento y desacoplamiento del software, hay que tener presente cuál es la situación de nuestro cliente. En este caso, el software de CaixaBankNow está desarrollado por una gran cantidad de equipos de trabajo distintos y, además, todo el código y estructura de la aplicación ya está diseñado.

Por consiguiente, la opción escogida es la más adecuada para este ámbito de trabajo ya que se añaden mejoras en el diseño del código, pero éstas no implican una reconstrucción total del software antiguo. En cambio, si el proyecto consistiera en el desarrollo de una nueva aplicación sí que sería válido e interesante plantear otro diseño del software como los comentados anteriormente.

Como conclusión, este proyecto es un gran reto que tiene como objetivo trabajar en mejorar la experiencia de usuario de los clientes de una entidad bancaria tan importante como lo es La Caixa. Se trabajará con unas tecnologías que, una vez analizadas con detenimiento, son las ideales para poder resolver el problema planteado. La ejecución del proyecto implicará, por una parte, un gran cambio respecto al paradigma actual a lo que software se refiere. Por otra parte supondrá que millones de usuarios en todo el mundo puedan realizar sus gestiones bancarias de una forma más rápida, cómoda, intuitiva, sencilla, ágil y visualmente más atractiva.

3. Alcance

En este apartado comentaremos el alcance de este proyecto de fin de grado. Se detallarán los objetivos a alcanzar y también se comentarán los requisitos, funcionales y no funcionales, que se buscan en el trabajo. En un último apartado se describirán los obstáculos y posibles riesgos que pueden surgir durante la realización del proyecto.

3.1. Objetivos del proyecto

En este apartado comentaré los objetivos de este proyecto de fin de grado. Los objetivos se dividen en dos categorías: objetivos principales y objetivos secundarios.

3.1.1. Objetivos principales

- Mejorar la experiencia de usuario de CaixaBankNow
- Renovar la operativa de Solicitar un anticipo de una factura
- Modificar pantallas de operativas de financiación y créditos documentarios

3.1.2. Objetivos secundarios

- Modularizar la estructura del código
- Aplicar patrones de diseño en la implementación de nuevas pantallas
- Generar ficheros de propiedades en herramientas internas de La Caixa para el control de proyectos de negocio
- Desplegar aplicaciones mediante herramientas de integración continua
- Cifrar datos de sesión los usuarios

3.2. Requisitos

3.2.1. Requisitos funcionales

Los requisitos funcionales del trabajo son las historias de usuario que me son asignadas por parte de la jefa de proyecto. Se engloban, principalmente, dentro del apartado de mejorar y modificar pantallas, estructura y la lógica asociada al proyecto de negocio llamado *Confirming*. Este proyecto engloba todo lo relacionado con las operativas que un usuario puede realizar con las facturas “pagadas” y “cobradas”. Dentro de este de proyecto de negocio se trabajará sobre la operativa de Solicitar el anticipo de facturas.

Esta operativa consiste en que un usuario de *CaixaBankNow* tenga la opción de poder anticipar el cobro de las facturas pendiente de sus proveedores. El anticipo conlleva el ingreso en la cuenta del usuario el importe de la factura. El importe que se abona es la diferencia entre el importe de la factura y una serie de comisiones e intereses.

A continuación, se detallan los requisitos funcionales de toda esta operativa. Los requisitos son solicitados por el cliente la Caixa y son extraídos de la herramienta *Confluence* de *CaixaBank*.

1. Seleccionar el proveedor del cual quiero anticipar las facturas
2. Consultar los anticipos de facturas de un proveedor
3. Seleccionar las facturas que deseo anticipar
4. Consultar cuánto tiempo queda hasta el vencimiento
5. Saber en qué fase estoy del proceso de anticipar facturas
6. Seleccionar una cuenta personal donde deseo que se realice el anticipo
7. Poder introducir el IBAN de una cuenta donde deseo que se realice el anticipo
8. Poder confirmar la operación de un modo seguro
9. Poder consultar un resumen de la operación
10. Poder saber si el anticipo se ha realizado con éxito

3.2.2. Requisitos no funcionales

- Seguridad

Al ser una aplicación de banca digital es indispensable que el principal requisito no funcional que se busque es la seguridad. Todos los traspasos de datos están cifrados, así como los datos de sesión del usuario. Esto es imprescindible para mantener la integridad de los datos y la confianza de los clientes, pero también para cumplir con la legislación correspondiente. En ella se encuentra la Ley Orgánica de Protección de Datos de Carácter Personal y la normativa europea PSD2.

- Mantenibilidad

Teniendo en cuenta la gran cantidad de gente que trabaja en Línea Oberta de La Caixa y la gran cantidad de modificaciones y mejoras que se buscan constantemente es muy importante que el software sea fácil de mantener.

- Usabilidad

Como he comentado, el principal objetivo a resolver del proyecto es mejorar la experiencia de usuario en la aplicación. Por lo tanto, todo el diseño tiene que estar pensado para que sea fácil de utilizar por cualquier tipo de usuario, sin que éste tenga que disponer de conocimientos previos.

- Modularidad

Dado que hace pocos meses se ha introducido el nuevo *framework FwkLO* es importante que todas las funcionalidades desarrolladas a partir de ahora sigan las pautas del framework. Una gran pieza de este cambio es la modularidad, se busca que las operativas sean independientes una de las otras y que se pueda tener el control de la funcionalidad, aunque se cambie una parte del código. Esto último también es posible gracias al uso de las comentadas Palancas.

3.3. Obstáculos y riesgos

Los principales riesgos que existen durante todo el desarrollo del proyecto surgen del hecho de que sea una aplicación de banca online de una entidad financiera. Es un *software* utilizado por millones de usuarios de todo el mundo y en el que se manejan datos muy importantes y críticos. Por lo tanto, es fundamental que se controlen todos los errores y excepciones que puedan aparecer. También es primordial hacer pruebas de todas las funcionalidades implementadas y evidenciar su correcto funcionamiento. Si no se realizan estas pruebas o si no se detecta un fallo en entornos de test, puede ser que la aplicación no funcione correctamente en producción y que una gran cantidad de usuarios se vean duramente afectados.

Los obstáculos encontrados en el desarrollo del proyecto han sido el impedimento de realizar ciertas acciones por no disponer de permisos suficientes o por demora en conseguirlos. Sin embargo, es algo completamente normal y lógico teniendo en cuenta en el contexto en el que se encuentra el trabajo. Es necesario que existan permisos para que no cualquier persona pueda tener acceso a alterar o modificar ciertas cosas de Línea Oberta. De esta manera se dota de seguridad a la aplicación y a su desarrollo.

4. Metodología y rigor

Todo el proceso de desarrollo se realiza siguiendo metodologías ágiles. Dentro de este tipo de metodologías, gran parte del proyecto está basado en el método *SCRUM*. Como veremos a continuación, se aplican los conceptos y procedimientos estudiados en la especialidad de *Software*. Minsait acostumbra a trabajar con este tipo de metodologías en este tipo de proyectos ya que, de esta manera, el cliente puede ver cada muy poco tiempo como avanza el progreso del producto final.

En este proyecto, este poco tiempo son dos semanas dado que es el periodo que dura cada *Sprint*. Al principio de cada *Sprint* o iteración, tiene lugar una reunión presencial con el cliente de La Caixa. En esta reunión se lleva a cabo una revisión anterior. Se determina qué es aquello que se ha logrado terminar y qué es lo que no, se realiza una demostración al cliente del producto y se analiza la iteración finalizada. Es también en esta reunión donde se produce la planificación del siguiente *Sprint*. Se construye un *Sprint Backlog*, que son aquellas historias de usuario que se realizarán en la siguiente iteración, a partir del *Product Backlog*, que son todas las funcionalidades que tendrá el producto final. Para la selección de las correspondientes historias de usuario, el equipo de desarrollo vota mediante *Planning Poker* la dificultad de cada historia de usuario. Finalmente, se escogen una cantidad de puntos de historia que concuerde con la velocidad del equipo, la media de puntos de historia que se logran realizar por *Sprint*.

A más a más, el equipo de trabajo realiza una reunión al final de cada *Sprint* para realizar la retrospectiva del mismo. Aquí, se revisa cómo fue la comunicación, el proceso, el uso de las herramientas, etc. Se analizan los errores que se cometieron, se identifican los temas que se realizaron con éxito y se proponen mejoras de cara al futuro.

Cada día, se realizan también reuniones telefónicas con el cliente y con el *Scrum Master*. En estas reuniones, llamadas *daily*s, cada miembro del equipo de desarrollo comenta que estuvo realizando el día anterior, los problemas que se encontró y explica en qué estará trabajando ese mismo día. En estas *daily*s también se hacen consultas al cliente sobre las dudas que surgen durante el desarrollo y se termina de detallar aquello que no quede claro de cada historia de usuario.

Gracias a seguir este tipo de metodología, se mantiene un contacto cercano y frecuente entre el cliente y el equipo de proyecto. De esta manera, es rápido resolver y completar la información que no quede clara, así como corregir todo aquello que el cliente desee modificar o añadir. Del mismo modo, desde la perspectiva del cliente es muy cómodo poder observar cada dos semanas mejoras y avances en el producto final.

Una vez asignada una tarea a cada miembro del equipo para la iteración, se siguen diversas pautas de trabajo. En primer lugar, se debe leer atentamente la historia de usuario y sus correspondientes criterios de aceptación. Para esto, La Caixa provee de una herramienta llamada *Confluence* que comentaremos en profundidad en el siguiente apartado. Cuando se detectan todos los requisitos se programa la funcionalidad descrita, tanto el diseño de la pantalla como la lógica asociada a ésta. Finalmente, se deben realizar *tests* de las clases java implementadas. Para los *tests* hay que tener siempre en cuenta que como mínimo deben de cubrir un 90% del código escrito, lo que en *JUnit* se conoce como *coverage*.

Existen cinco entornos de desarrollo: Local, Integración, Preproducción, Simulación y Producción. Hasta el momento descrito anteriormente, todas las pruebas se realizan en local. Esto es debido a que no es posible subir a nada al siguiente entorno, es decir Integración, si no se han realizado con éxito los *tests* explicados.

En el momento en que la historia de usuario requerida funciona correctamente en el entorno de Integración, es necesario realizar evidencias de dicha funcionalidad. Las evidencias es documentar de manera gráfica que, efectivamente, se ha completado la implementación y que ésta cumple con todos los criterios de aceptación.

Finalmente, cuando se cumple correctamente el proceso descrito, se marca la correspondiente historia de usuario como completada. Sin embargo, es posible que en determinadas ocasiones no suceda esto antes de acabar el *Sprint*. De ser así, se actúa de la misma manera que se ha explicado en la universidad, las historias de usuario incompletas se traspasan al *Backlog* del siguiente *Sprint*. No obstante, si por el contrario se logra completar el trabajo antes de la finalización de la iteración, se empieza con una nueva historia de usuario pensada para el posterior *Sprint*. Éstos son los llamados *Nice To Have*, funcionalidades que se desean implementar pero no son necesarias para la actual iteración del proyecto.

Debido a la pandemia del virus Covid-19 la metodología expuesta anteriormente sufre ligeros cambios. Debido a que a partir del 11 de marzo todo el equipo de trabajo pasó a trabajar de forma telemática, todas las reuniones en adelante fueron mediante llamada. Por lo que respecta al resto de la metodología de trabajo no es necesario realizar ninguna modificación.

4.1. Herramientas de seguimiento

En este apartado se comentarán las principales herramientas de seguimiento que se utilizan por el equipo de trabajo para el desarrollo del proyecto. Mediante estas herramientas se ha podido llevar un control de las tareas, tanto de su estado como la descripción de requisitos.

- Jira

Software creado por *Atlassian* para la administración de tareas de un proyecto, el seguimiento de errores e incidencias y para la gestión operativa los mismos [7]. Desde esta herramienta es posible seguir el avance del proyecto y de sus respectivas *Sprints* con comodidad.

- Confluence

Software creado por *Atlassian* [8]. En él se documentan las historias de usuario junto con sus correspondientes criterios de aceptación. También hay disponible documentación sobre el *framework FwkLO*, sobre la realización de los *test* y un manual de buenas prácticas.

5. Duración del proyecto

En este apartado se define la duración del proyecto, desde su inicio hasta la fecha prevista de entrega de la memoria, y el tiempo dedicado semanalmente. En primer lugar, al ser un trabajo de final de grado que se realiza en modalidad B, con prácticas curriculares en empresa, existe un mínimo de carga de trabajo que son 735 horas [9]. Esto se ve reflejado en el periodo de las prácticas, ya que su inicio es el día 10 de febrero de 2020 y el final está previsto que sea el día 19 de junio de 2020. La dedicación semanal al desarrollo del proyecto es de 40 horas. Por lo que en las 19 semanas que dura la estancia de prácticas en la empresa se alcanza el mínimo de horas fijado por la facultad.

Además, se deben tener en cuenta las horas de gestión y de documentación del proyecto que veremos más adelante en la descripción de tareas. La lectura del proyecto está prevista para el turno de junio de 2020.

6. Planificación temporal

Al ser un proyecto desarrollado con metodologías ágiles, en concreto utilizando *Scrum*, se divide el trabajo en iteraciones llamadas *Sprints*. Cada *Sprint* tiene una duración de dos semanas y empieza con una reunión de planificación y finaliza con una retrospectiva donde se valora qué mejorar de cara a las siguientes dos semanas.

Los *Sprints* se dividirán en tareas más concretas de manera que sea más sencillo realizar una estimación de su duración, así como una explicación con más detalle.

6.1. Definición y estimación de las tareas

6.1.1. Gestión del proyecto:

Realizar una planificación del proyecto. Documentar el alcance y estimación previstos. Calcular un presupuesto del total del proyecto e identificar los posibles riesgos y la gestión de los mismos. Esta tarea incluye también la creación de un informe de sostenibilidad del proyecto.

- Reunión con profesores de la facultad: Explicación de gestión del proyecto y aclaración de dudas.

Estimación: 2 horas

Dependencias: -

Recursos humanos: Profesorado de la facultad

Recursos materiales: Documentación

- Reunión con jefa de proyecto: Determinar alcance del proyecto. Identificar el problema a resolver y explicación de metodología de trabajo. Evaluación de riesgos

Estimación: 3 horas

Dependencias: -

Recursos humanos: Jefa de proyecto

Recursos materiales: Ordenador

- Elaboración de documentación: Creación y corrección de la documentación necesaria para la adecuada gestión del proyecto.

Estimación: 70 horas

Dependencias: -

Recursos humanos: Profesor de la facultad y ponente

Recursos materiales: Ordenador

6.1.2. Sprints

Al ser un proyecto realizado con *Scrum* y que se adapta a los requerimientos del cliente en función del tiempo, se detallaran las tareas que se comparten en cada *Sprint*. No es posible precisar que requisito funcional se desarrollará en cada iteración.

- Sprint planning: Reunión entre el equipo de desarrollo, el cliente y el *Scrum Master*. Se realiza siempre al empezar una nueva iteración. Se construye el *Sprint backlog* y se cada historia de usuario a un miembro del equipo.

Estimación: 2 horas

Dependencias: Restrospective meeting del Sprint anterior

Recursos humanos: Equipo de desarrollo, cliente y *Scrum Master*

Recursos materiales: Ordenador

- Daily meeintg: Reunión diaria con el cliente para evaluar el proceso de desarrollo y para la consulta de posibles dudas.

Estimación: 0,5 horas

Dependencias: *Daily meeting* del día anterior

Recursos humanos: Equipo de desarrollo, cliente y *Scrum Master*

Recursos materiales: Ordenador, teléfono

- Implementación requisito funcional: Desarrollo de la historia de usuario asignada para la iteración. Se escribe el código necesario teniendo en cuenta el *framework* utilizado y los requisitos no funcionales del proyecto.

Estimación: 45 horas

Dependencias: *Sprint planning*

Recursos humanos: Equipo de desarrollo

Recursos materiales: Ordenador

- Pruebas e integración: Se realizan los tests necesarios para comprobar que el código implementado es correcto. Posteriormente se integra el nuevo requisito funcional con el resto del proyecto.

Estimación: 20 horas

Dependencias: Implementación requisito funcional

Recursos humanos: Equipo de desarrollo

Recursos materiales: Ordenador

- Retrospective meeting: Reunión del equipo de trabajo para analizar los errores cometidos y para corregirlos. Se identifican también aquello que se ha realizado de forma correcta para potenciarlo.

Estimación: 1 hora

Dependencias: -

Recursos humanos: Equipo de desarrollo

Recursos materiales: Ordenador

6.1.3. Seguimiento y control del proyecto

- Reuniones de seguimiento con el ponente del proyecto para el control y corrección del proyecto.

Estimación: 5 horas

Dependencias: -

Recursos humanos: Ponente

Recursos materiales: Ordenador

6.1.4. Documentación del proyecto

- Elaboración de la memoria final del proyecto: Incluye la documentación de la parte técnica del trabajo, así como la adaptación y mejora de la documentación previa. También contiene las reuniones con el ponente.

Estimación:	60 horas
Dependencias:	Gestión del proyecto
Recursos humanos:	Ponente
Recursos materiales:	Ordenador

6.2. Resumen de tareas

En la siguiente tabla se muestra un resumen de las tareas definidas previamente y la duración individual de cada una de ellas y la duración total del proyecto. Como resultado de esta planificación temporal, obtenemos que el proyecto tendrá una duración de 770 horas. Se han omitido el detalle de las tareas de los últimos cinco *Sprints* dado que es exactamente igual a los anteriores.

<i>Tarea</i>		<i>Duración tarea (horas)</i>	<i>Duración total (horas)</i>
1. Gestión del proyecto	1.1 Reunión FIB	2	75
	1.2 Reunión Jefa proyecto	3	
	1.3 Documentación	70	
2.1 Sprint 1	2.1.1 Sprint Planning	4	70
	2.1.2 Implementación	45	
	2.1.3 Tests e integración	20	
	2.1.4 Retrospective meeting	1	
2.2 Sprint 2	2.1.1 Sprint Planning	4	70
	2.2.2 Implementación	45	
	2.2.3 Tests e integración	20	
	2.2.4 Retrospective meeting	1	
2.3 Sprint 3	2.3.1 Sprint Planning	4	70
	2.3.2 Implementación	45	
	2.3.3 Tests e integración	20	
	2.3.4 Retrospective meeting	1	
2.4 Sprint 4	2.4.1 Sprint Planning	4	70
	2.4.2 Implementación	45	
	2.4.3 Tests e integración	20	
	2.4.4 Retrospective meeting	1	
2.5 Sprint 5			70
2.6 Sprint 6			70
2.7 Sprint 7			70
2.8 Sprint 8			70
2.9 Sprint 9			70
3. Seguimiento y control del proyecto	3.1 Reunión con ponente 1	1	5
	3.2 Reunión con ponente 1	2	
	3.3 Reunión con ponente 1	2	
4. Documentación del proyecto	4.1 Elaboración de memoria	60	60
			770

Tabla 1. Resumen de tareas

6.3. Diagrama de Gantt

El diagrama de *Gantt* de todo el desarrollo del proyecto se adjunta como el anexo A al final de este documento. El eje temporal se inicia el 10 de febrero de 2020 que coincide con el inicio del periodo de convenio de cooperación educativa con la empresa *Minsait* y finaliza también con el fin de este convenio el 19 de junio de 2020.

Al ser un proyecto realizado con metodologías ágil, se ha optado por integrar los *Sprints* en la propia lista de tareas. Dentro de cada iteración se encuentran las tareas comentadas en el anterior apartado. Además, se han añadido una estimación de la fecha de las reuniones de seguimiento con el ponente para el control del proyecto. La herramienta con la que se ha elaborado el diagrama es *TeamGantt* [10].

7. Gestión del riesgo y planes alternativos

En este apartado se comentan los obstáculos y riesgos identificados que pueden surgir en este proyecto. Se especifica también la probabilidad de que puedan manifestarse, así como el grado de impacto en el trabajo. También se añade el plan alternativo estudiado y planteado para tratar de mitigar el daño que puede llegar a causar en el desarrollo.

- Planificación temporal incorrecta

Probabilidad	Baja
Impacto	Leve

Plan alternativo:

Al utilizar metodologías ágiles, ya se tiene en cuenta siempre la posibilidad de que algún elemento pueda requerir más tiempo del inicialmente planteado. Simplemente se trasladaría ese requisito al siguiente *Sprint*. Afectaría de forma leve ya que lo haría sería retrasar un corto periodo de tiempo el proyecto. El coste adicional sería el coste de personal extra por tarea.

- Demora en obtención de permisos

Probabilidad	Alta
Impacto	Moderado

Plan alternativo:

Si se retrasa la obtención de un permiso necesario para realizar una acción tanto como para que afecte la entrega de la misma, siempre puede otro miembro del equipo realizar dicha entrega en mi nombre. De esta manera, dado que ese miembro seguro que tiene dicho permiso necesario, se podrá realizar con éxito y a tiempo la entrega. No presenta, por tanto, coste adicional.

- Error grave en el requisito implementado

Probabilidad	Baja
Impacto	Grave

Plan alternativo:

Para evitar este riesgo que afectaría gravemente a la aplicación y a sus usuarios existen diferentes entornos de prueba y de simulación. En estos entornos se realizan numerosas pruebas y comprobaciones por parte de diferentes equipos de trabajo precisamente para que algo así no pueda llegar a suceder, por lo que no habría que dedicar recursos adicionales. Si el error se detectara en una fase tardía de las pruebas podría llegar a afectar a la subida al entorno de producción, pero no afectaría a la duración total del proyecto.

- Pandemia global Coronavirus COVID-19

Probabilidad	Alta
Impacto	Moderado

Plan alternativo:

Debido a la expansión del virus *COVID-19*, declarada como pandemia global por la *OMS*, la empresa Minsait decidió cancelar la asistencia a las oficinas. Sin embargo, esto no produjo significativos problemas para el desarrollo del proyecto. Esto es debido a que Indra tiene ya desde hace tiempo un programa de teletrabajo llamado *EasyWorking*. Por lo tanto, toda planificación y el desarrollo continúan de la misma manera. Esta situación lo que implica es que, durante un tiempo, no se realizan las reuniones ni el propio trabajo de forma presencial. Este plan no presenta un coste añadido puesto que únicamente cambia el lugar de trabajo.

8. Presupuesto

8.1. Identificación de costes

8.1.1. Costes de personal por actividad

En primer lugar, es necesario determinar el sueldo que tiene cada perfil de trabajador por hora. Para ello, se ha recopilado en esta tabla los sueldos orientativos que Indra proporciona a sus distintos empleados [11]. Comentar que estos serían los costes de personal por parte de la empresa, es decir, ya se ha incluido la contribución a la seguridad social.

<i>Perfil</i>	<i>Coste por hora (€)</i>
Jefe de proyecto	18,5
Analista del software	17,7
Desarrollador frontend	15,3
Tester	16

Tabla 2. Coste por hora según tipo perfil

Ahora que tenemos detallado cual es el coste por hora de cada perfil que participará en el proyecto, es momento de calcular los costes de cada tarea. En la siguiente tabla se muestra, por cada tarea, cuantas horas se dedican según el rol desempeñado. Se añade también el coste de cada tarea y el coste total de personal por actividad (CPA). Estos valores son el resultado de multiplicar las horas que se dedican de cada rol por el coste por hora del correspondiente perfil.

Tarea	Horas de trabajo según perfil				Coste (€)
	Jefe de proyecto	Analista de software	Desarrollador frontend	Tester	
Gestión proyecto	75				1.388
Sprint 1		5	45	20	1.097
Sprint 2		5	45	20	1.097
Sprint 3		5	45	20	1.097
Sprint 4		5	45	20	1.097
Sprint 5		5	45	20	1.097
Sprint 6		5	45	20	1.097
Sprint 7		5	45	20	1.097
Sprint 8		5	45	20	1.097
Sprint 9		5	45	20	1.097
Seguimiento proyecto	5				93
Elaboración memoria	60				1.110
Total	140	45	405	180	12.464
	770				

Tabla 3. Coste de personal por actividad

8.1.2. Costes genéricos

En este apartado se computan los costes genéricos del proyecto. Para calcular el coste del espacio de trabajo se ha estimado a partir de una oficina de *coworking* en la misma zona donde se encuentra las oficinas de Indra [12]. Para el ordenador, en cambio, se computa el coste de amortización calculado con la siguiente fórmula:

$$\left(\frac{\text{Coste equipo } \text{€}}{\text{Años vida útil} \cdot 220 \text{ días laborables/año} \cdot 8 \text{ horas/día}} \right) \cdot 770 \text{ horas TFG}$$

<i>Elemento</i>	<i>Unidades</i>	<i>Vida útil</i>	<i>Coste (€)</i>	<i>Amortización (€)</i>
Ordenador Portátil	1	4 años	1.450	159
Transporte T-Jove	2	3 meses	200	-
Conexión internet	1	4 meses	240	--
Espacio de trabajo	1	4 meses	752	-
			1.351	

Tabla 4. Costes genéricos

8.1.3. Contingencias

Como en todo proyecto, es necesario preparar una partida de contingencias. Este fondo económico contempla la posibilidad de que surjan imprevistos no anticipados. De esta manera, se permite minimizar el impacto de errores provocados por falta de información o por descuidos en el momento de la planificación.

En este caso se ha optado por un nivel de contingencia del 15 % del coste del presupuesto calculado hasta ahora. Es decir, sobre la suma del total de costes de personal por actividad (CPA) y del total de costes genéricos (CG).

$$0.15 \cdot (\text{Total CPA} + \text{Total CG}) = 0,15 \cdot (12.464 + 1.351) = 2.072,25 \text{ €}$$

Por lo tanto, el coste de contingencias del proyecto será de 2.073 Euros.

8.1.4. Imprevistos

Contaremos como coste de imprevistos la posibilidad de que se necesite un Sprint extra para finalizar el desarrollo del proyecto. Aunque muchos requisitos funcionales sean críticos y puedan surgir errores, al estar rodeado de un equipo con experiencia, se fija una probabilidad baja: un 25%. Por lo tanto, calcularemos este coste de imprevisto como el coste de personal de un Sprint en proporción a su probabilidad de ocurrencia.

$$1097 \text{ €} * 0,25 = 275 \text{ €}$$

El coste por imprevistos será de 275 €. Comentar también que, en el caso de que no surjan estos problemas, se destinarán los costes de contingencias y de imprevistos a cubrir el coste de personal necesario para que se realicen unas pruebas exhaustivas para asegurar el correcto funcionamiento de la aplicación.

8.1.5. Costes totales

El presupuesto total del proyecto es de 16.163 €. En este coste, como hemos visto, se recogen los costes de personal por actividad, costes genéricos, costes de contingencias y de imprevistos.

<i>Costes</i>	<i>Importe (€)</i>
Personal por actividad	12.464
Genéricos	1.351
Contingencias	2.073
Imprevistos	275
Total presupuesto	16.163

Tabla 5. Costes totales

8.2. Control de gestión

Para realizar una buena gestión del proyecto es necesario mantener un control exhaustivo del presupuesto elaborado. Por esta razón, conforme se vayan realizando las tareas planificadas y se vaya avanzando el desarrollo del proyecto se actualizará el presupuesto original.

Se calcularán las desviaciones entre los valores reales y los valores estimados en la fase de planificación del proyecto. Para ello se utilizarán los siguientes indicadores de desviación.

$$\text{Desviación CPA} = (\text{Coste por hora estimado} - \text{Coste por hora real}) * \text{Horas reales}$$

$$\text{Desviación horas por tarea} = (\text{Horas estimadas} - \text{Horas reales}) * \text{Coste estimado}$$

$$\text{Desviación CG} = \text{CG estimado} - \text{CG real}$$

$$\text{Desviación CG} = \text{CG estimado} - \text{CG real}$$

$$\text{Desviación coste imprevistos} = \text{Coste imprevistos estimado} - \text{Coste imprevisto real}$$

$$\text{Desviación coste total} = \text{Coste estimado total} - \text{Coste real total}$$

Debido a la pandemia del Covid-19 se pasó a trabajar de forma telemática por lo que se produjo un cambio en los costes genéricos respecto a los costes genéricos estimados en la fase de planificación. Estos cambios afectaron a los siguientes conceptos: Transporte T-Jove, que solo fue necesaria una unidad, y al espacio temporal de trabajo, que se redujo de cuatro meses a un mes.

Recalculamos, a continuación, los costes genéricos del proyecto.

<i>Elemento</i>	<i>Unidades</i>	<i>Vida útil</i>	<i>Coste (€)</i>	<i>Amortización (€)</i>
Ordenador Portátil	1	4 años	1.450	159
Transporte T-Jove	1	3 meses	100	-
Conexión internet	1	4 meses	240	--
Espacio de trabajo	1	1 mes	188	-
			687	

Tabla 6. Costes genéricos recalculados

El importe de los costes genéricos reales fue de un total de 687 Euros. Por lo tanto, se produjo una desviación de 664 Euros.

$$\text{Desviación CG} = \text{CG estimado} - \text{CG real} = 1351 \text{ €} - 687 \text{ €} = 664 \text{ €}$$

No se han detectado ninguna desviación más de costes. Por lo tanto, la desviación del coste total es de 664 €.

9. Sostenibilidad

9.1. Autoevaluación

Para poder articular este apartado he realizado la encuesta propuesta por parte del equipo de profesores de la asignatura de gestión de proyectos. Es muy interesante ponerse uno mismo a prueba y evaluar realmente el conocimiento que se tiene sobre un tema específico. La autoevaluación sobre la sostenibilidad es relevante porque es una cuestión que normalmente no se conoce en profundidad.

Antes de realizar dicha encuesta, diseñada por el proyecto EDINSOT, creía tener conocimientos a un nivel global de todos los ámbitos donde tiene sentido hablar de sostenibilidad. Sin embargo, después de acabar, he podido darme cuenta realmente del nivel de conocimientos que dispongo sobre este tema. La verdad es que sí que era consciente del significado de la mayoría de conceptos que aparecían. No obstante, se presentaban distintas ideas y problemáticas de las cuales no era consciente de la relevancia que tienen en un proyecto de desarrollo de una aplicación como lo es este. La dimensión ambiental es quizás el apartado que menos se tiene en consideración al inicio de un proyecto informático. Precisamente por eso se debe prestar más atención al posible impacto medioambiental que pueda provocar la aplicación de dicho proyecto.

Como conclusión, decir que a partir de este momento y de cara al futuro tendré en cuenta los diferentes ámbitos de la sostenibilidad que existen y cómo poder tratarlos y analizarlos de la forma correcta. Espero poder poner esto mismo en práctica ya en este proyecto. De esta manera, me resultará más sencillo en próximos trabajos comprender los problemas que tengan relación con la sostenibilidad y como dar solución a ellos.

9.2. Dimensión económica

El coste del presupuesto de este proyecto que se ha estimado es un coste realista. Es un proyecto importante y ambicioso que tiene como cliente a una gran entidad financiera como es La Caixa.

Esta solución mejorará la eficiencia de la aplicación. Por lo tanto, no serán necesarios destinar, por parte del cliente, tantos recursos a mantener un software antiguo y con defectos. Además, con todo el plan de renovación de la aplicación, La Caixa, espera obtener un retorno de la inversión al disponer de una mejor, más atractiva, más útil y más eficiente plataforma.

9.3. Dimensión ambiental

El impacto negativo ambiental de este proyecto es nulo. Esto es debido a que se está modificando una aplicación web que actualmente ya existe y que ya está en pleno funcionamiento en todo el mundo. Por lo que se están reutilizando los servidores donde está alojada actualmente la aplicación. Sin embargo, sí que se ha planteado el impacto ambiental del desarrollo del proyecto. En cuanto a este aspecto, podemos afirmar que también es prácticamente nulo. Los motivos son porque gran parte del período de desarrollo se ha realizado desde casa sin efectuar ningún tipo de desplazamiento que pudiera provocar un impacto medioambiental. Además, desde la propia empresa Minsait, se reutilizan todos los ordenadores portátiles.

9.4. Dimensión social

En cuanto a la dimensión social del proyecto, personalmente me aportará una gran cantidad de experiencia y de conocimientos técnicos. Además, gracias a realizarlo en Minsait, podré aprender su metodología y forma de trabajar en equipo. También he aprendido nuevo conocimiento sobre la gestión, la planificación y presupuesto de proyectos con la parte de GEP.

Por otro lado, este trabajo beneficia en gran medida a los usuarios finales de la aplicación CaixaBankNow. Podrán disfrutar de nuevas funcionalidades para un mejor desempeño de sus operaciones bancarias, así como una mejor experiencia en la plataforma.

10. Arquitectura y diseño

10.1. Patrones arquitectónicos

Modelo vista controlador

La propuesta decidida para dar solución al problema de qué tipo de arquitectura seguir ha sido el modelo-vista-controlador. Este patrón arquitectónico que separa la lógica de la aplicación y los datos de su representación y de la interfaz de usuario. Es un tipo de arquitectura muy sólida y madura que tiene buen rendimiento en todo tipo de plataformas y de lenguajes.

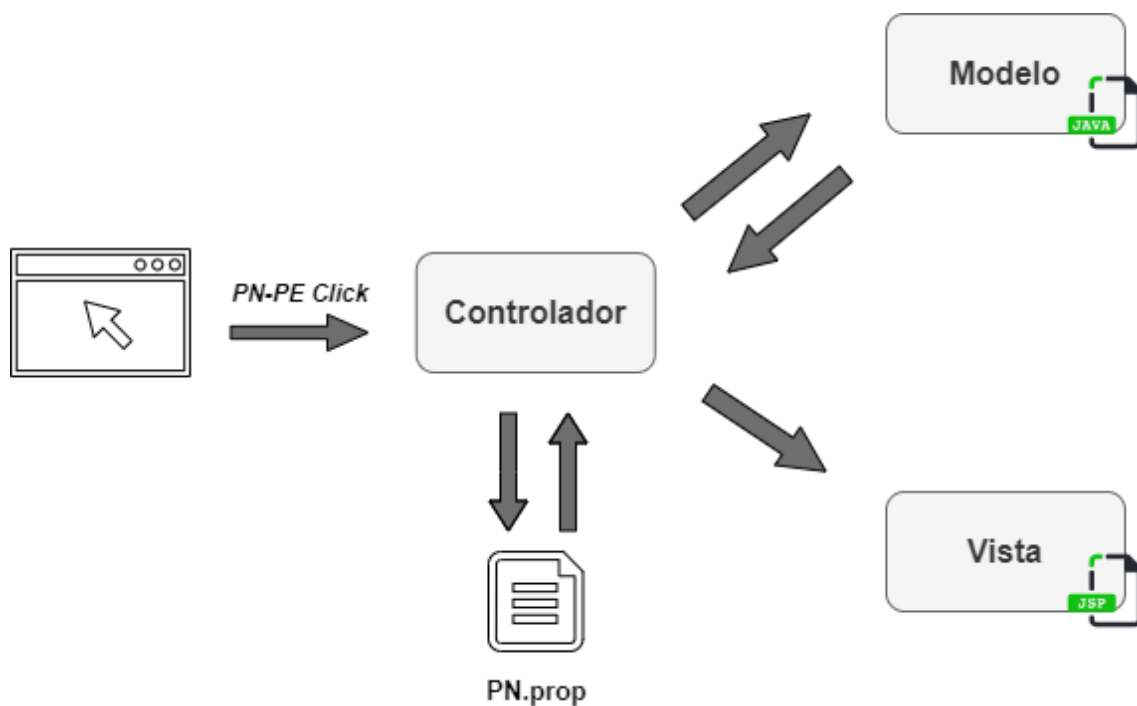


Figura 1. Modelo Vista Controlador

El proceso de una petición se inicia cuando el usuario realiza un *clic* en la interfaz de la aplicación. Este *clic* podría ser, por ejemplo, avanzar a la confirmación de un pago a realizar. En ese momento, se enviaría la información de la acción que se desea realizar y el identificador de la sesión del usuario.

Desde el servidor se recibiría la URL con esta información y se redirigiría al proyecto de negocio correspondiente. El controlador es el encargado de este proceso. Gracias al

identificador de sesión de usuario y a la información de la acción en la URL es posible obtener todos los datos necesarios para redirigir la petición al modelo.

En el modelo manipula los datos y aplica la lógica necesaria para cada petición que se realiza. También es el encargado de realizar consultas a la base de datos y a diferentes servicios dependiendo, nuevamente, de la petición que se ha efectuado. Una vez se han obtenido los datos y modificado se envía al controlador la información que va a necesitar la vista.

Seguidamente, el controlador carga la vista correspondiente a la petición realizada y le traspasa los datos obtenidos de la capa del modelo. Esta vista será, finalmente, transformada en HTML y enviada al cliente que realizó la petición inicial.

En el siguiente diagrama de secuencia podemos ver con más detalle el flujo de ejecución desde que el usuario realiza una petición en la interfaz de la aplicación mediante un navegador hasta que el sistema le retorna una nueva vista. Podemos ver las llamadas que se realizan entre los diferentes componentes de la aplicación.

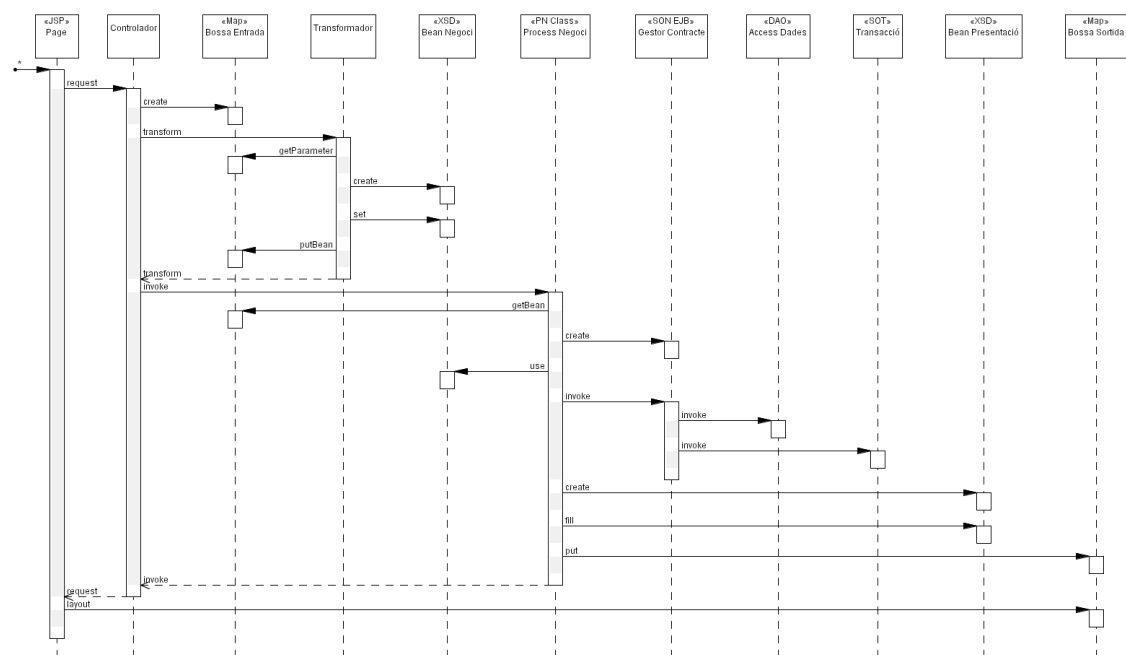


Figura 2. Diagrama de secuencia - Petición

10.2. Patrones de diseño

Data transfer Object

Este patrón consiste en el uso de un objeto de transferencia de datos (data transfer object, DTO). Este objeto sirve para transportar datos entre distintos procesos. La motivación de su uso tiene como objetivo reducir la complejidad y reducir de la comunicación entre procesos.

En este proyecto utilizamos un DTO para transportar la información obtenida en el proceso que depende de la lógica del proyecto de negocio y de la petición solicitada al proceso de construcción de la vista.

Business Delegate

Utilizamos el patrón de diseño Business Delegate para reducir el acoplamiento entre los clientes de la capa de presentación y los servicios de negocio. Este patrón nos permite reducir ese acoplamiento ocultando los detalles de la implementación del servicio de negocio. Esto es importante ya que diferentes clientes, dispositivos, clientes Web, y programas, necesitan acceder a los servicios de negocio de CaixaBank.

Inyección de dependencias

Este patrón, también conocido como de inversión de control tiene como finalidad conseguir un código más desacoplado, que nos facilitará el trabajo a la hora de realizar los tests. Además, nos permite cambiar, si fuera necesario, partes del sistema más fácilmente. Consiste en poder pasar (inyectar) las dependencias cuando sea necesario en lugar de inicializar las dependencias dentro de la clase receptora.

Siguiendo el patrón de inyección de dependencias los componentes de la aplicación declaran sus dependencias, pero no se encargan de conseguirlas. El responsable es el framework FwkLO que es el encargado de conseguir e inyectar las dependencias a los objetos.

Factoría

El patrón de diseño Factoría nos permite la creación de determinados objetos ocultando sus respectivos detalles de creación. Se accede a través de una interface común para todos los tipos de objetos, de esta manera, éstos pueden variar sin afectar a la forma en la que el cliente interactúa con ellos. Con este patrón se delega la responsabilidad de la creación de objetos en la clase Factoría.

En este proyecto se ha utilizado este patrón en la llamada *BeanFactory*. Esta factoría se encarga de crear el *Java Bean* adecuado en cada ocasión. Estos objetos creados a partir de la factoría sirven para la creación de las vistas de la aplicación. Contienen la implementación de los componentes que conforman una vista: títulos, tablas, desplegables, etc.

Gracias a este patrón, podemos delegar la creación de los distintos tipos de componentes en la factoría y, simplemente, trabajar con un objeto título.

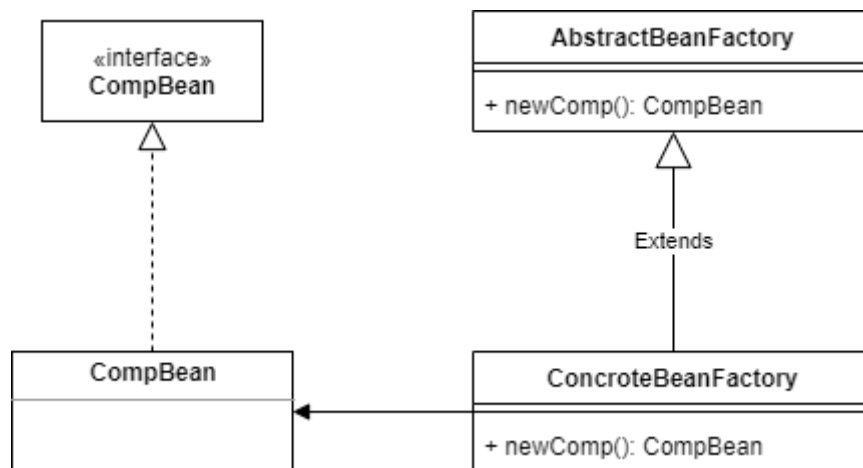


Figura 3. Patrón factoría

Singleton

El patrón de diseño singleton que pertenece a la categoría de patrones creacionales. Este patrón permite que solo exista una instancia de una clase a la vez. Esto es posible dado que el constructor de la clase en cuestión se convierte en privado. De esta manera, cuando

se requiere el objeto se utiliza la función *getInstance()* que siempre devuelve la misma instancia.

En concreto, se ha utilizado este patrón para la instanciación de la clase *gLogs*. Una clase para poder escribir *logs* o registros de la aplicación.

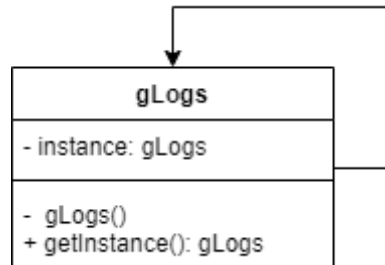


Figura 4. Patrón singleton

10.3. Herramientas utilizadas

- Git

Sistema de control de versiones de código abierto. Tenemos disponible una rama para cada proyecto de negocio (PN). Todo aquel miembro que quiera crear una rama de uno de estos proyectos, deberá hacerlo con el formato de nombre siguiente: *XXX_nombre* de usuario. De esta manera se puede estructurar mejor los repositorios y el equipo se puede organizar mejor.

- Mockito

Es un framework para el desarrollo de tests. Permite la creación de objetos dinámicos (*mock object*) que simulan parte del comportamiento de una clase. Está pensado para facilitar el desarrollo guiado por pruebas (*test-driven development*). Esta práctica consiste en escribir primero los test para realizar una refactorización posterior.

- Jenkins

Herramienta que permite la práctica de la integración continua del desarrollo. Existe un atributo en el fichero *pom* de cada proyecto java que, de estar en valor *true*, genera un paquete en *Jenkins*. Desde aquí es posible comprobar que los proyectos compilan y que

pasan con éxito los *tests* y superan el 90% de *coverage*. Cada día, hay tres subidas automáticas al entorno de Integración.

- JUnit

Librería desarrollada para poder probar el funcionamiento de las clases y métodos que componen la aplicación. Todos los *test* deben pasar como mínimo el 90% de *coverage* del código.

- SonarQube

Es una plataforma desarrollada en *Java* que nos permite realizar análisis de código con diferentes herramientas de forma automatizada [13]. Se utiliza para inspeccionar la calidad del código y detectar errores en el código y vulnerabilidades. Es ideal que no aparezca ningún aviso de posible fallo del *software* y, de ser así, se trata de corregir.

- Subversion

Herramienta para el control de versiones de los proyectos. Es un sistema centralizado en el que se suben las funcionalidades que han sido ya testeadas previamente. Es aquí donde se realizan las subidas de código que, posteriormente, se integraran en algún entorno de desarrollo.

11. Implementación

A continuación, se detallará el proceso de implementación del proyecto. Dado que, como hemos explicado anteriormente, se han utilizado metodologías ágiles para el desarrollado, este apartado será dividido en Sprints. De esta manera es posible comprender como ha avanzado el proyecto en cada iteración y así tener una visión completa de la fase de implementación.

En esta sección del proyecto explicaremos también de qué manera afecta el trabajo a los usuarios finales, cuáles son las mejoras y beneficios que recibirán al completar el plan de renovación. Para reflejar este progreso se comparará la versión actual de la aplicación CaixaBankNow con la renovación realizada en este proyecto. Como se ha comentado al principio del documento, se tiene que conseguir que la interfaz ayude y facilite la vida al usuario. No basta con que sea atractiva y funcional.

Gran parte del proyecto consiste en la renovación de la operativa de Solicitar anticipos de mis facturas. Esta operativa está disponible para los clientes Empresa de CaixaBankNow dentro del apartado de Financiación y en la sección de Confirming de proveedor. Con este gran plan de renovación de la aplicación La Caixa espera obtener un retorno de la inversión y situarse como una de las mejores plataformas de banca online del mercado.

“Por cada dólar invertido en UX Design, el rendimiento es de 100 dólares: las empresas pueden usar los beneficios del diseño del primer usuario para mejorar su rendimiento de la inversión, conectarse con su público objetivo y dejar espacio para el crecimiento.” [14]

Para situarnos correctamente antes de empezar a describir cada Sprint se presentan las figuras 5 y 6. Estas figuras muestran cómo acceder a esta operativa dentro de la aplicación de banca online de CaixaBank en modalidad de empresas.

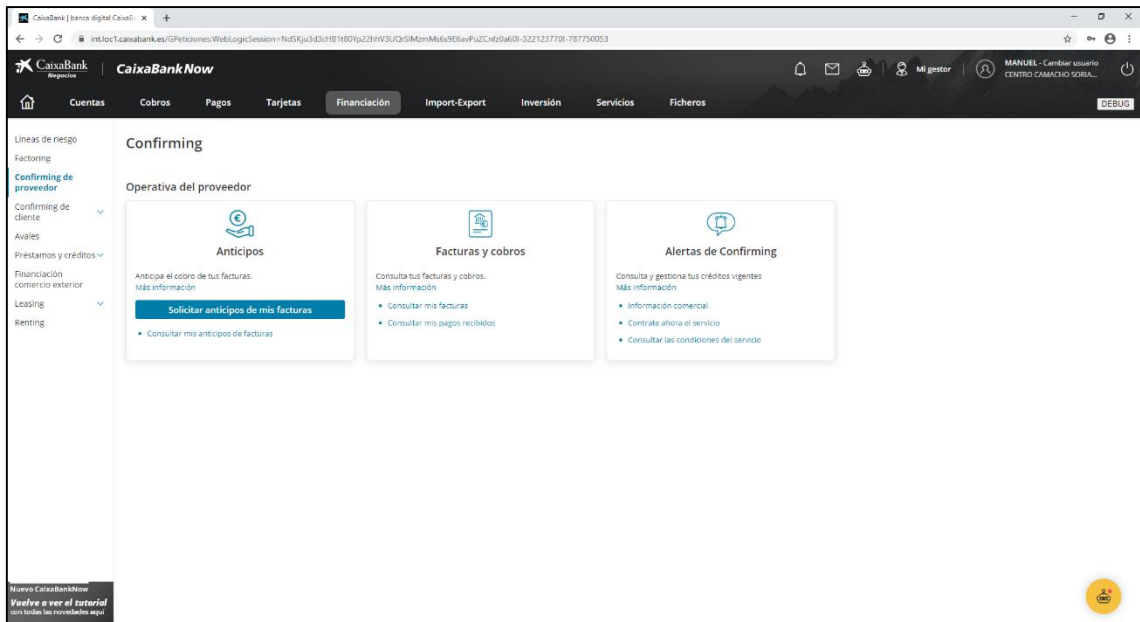


Figura 5. Confirming CaixaBankNow

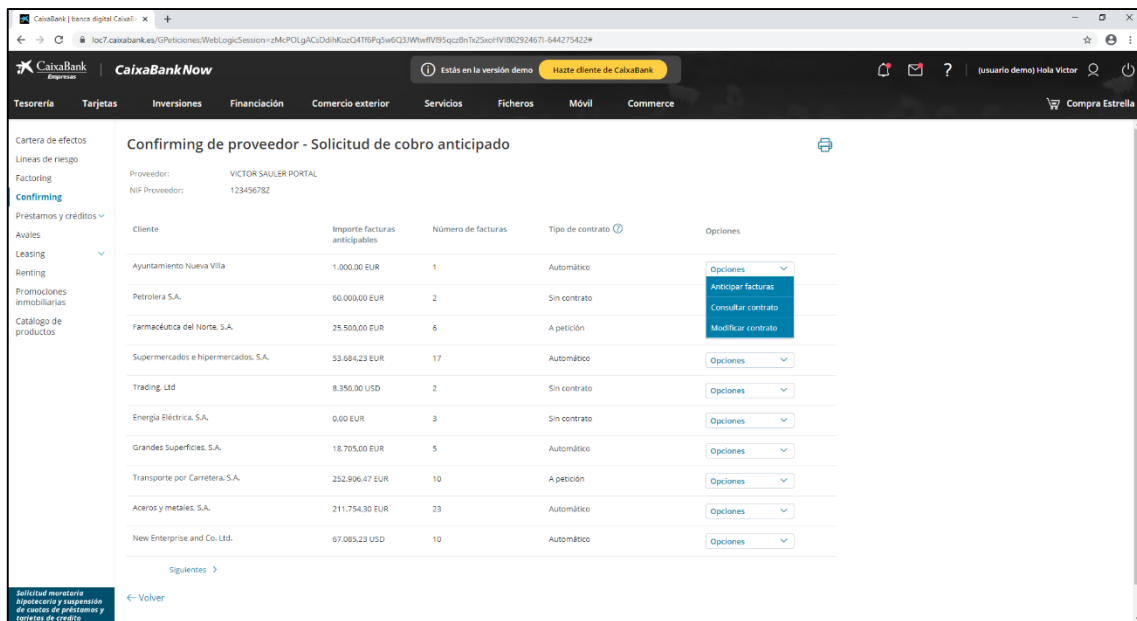


Figura 6. Solicitud de cobro anticipado - Selección de cliente

Una vez el usuario hace clic sobre la opción de “Solicitar anticipo de mis facturas”, se abre una nueva vista donde aparecen una tabla con todos los clientes de los que disponga dicho usuario. Seleccionando la opción de “Anticipar facturas” se inicia el proceso de la principal operativa que vamos a tratar durante este proyecto.

Cabe recalcar que todos los datos que aparecen en las capturas de pantalla de la aplicación CaixaBankNow son ficticios y pertenecen a usuarios de demostración.

11.1. Sprints

Sprint 1 y 2

El objetivo del primer Sprint fue montar el entorno de desarrollo local y familiarizarse con él. Como hemos dicho, la gran parte del código es en Java y se ha trabajado con el entorno de desarrollo integrado (IDE) llamado Eclipse.

También se realizó en esta primera etapa la instalación y configuración de un servidor local Weblogic para facilitar y agilizar el desarrollo individual de las tareas.

Asimismo, en estos primeros Sprints de inicio del proyecto dediqué gran parte del tiempo a aprender y analizar la estructura del código y, por supuesto, de FwkLO - el nuevo framework que se utilizará en la aplicación.

Conjuntamente con el estudio de la parte más técnica del proyecto, también fue necesario invertir tiempo y esfuerzo en el aprendizaje de las herramientas complementarias de CaixaBankNow.

Además, se tuvieron que pedir los permisos necesarios a Silk para poder acceder a las distintas herramientas de desarrollo y mantenimiento de CaixaBankNow. Dichos permisos incluían el acceso a todo lo relacionado con el código: como el repositorio, la plataforma SonarQube, el servidor Jenkins; así como también el acceso al Confluence, herramientas de literales, herramienta de despliegue, al Jira y el acceso al disco compartido en red.

Estas autorizaciones permiten el acceso a las distintas plataformas y herramientas con el objetivo de restringir la entrada a toda persona que no sea del equipo de desarrollo o de negocio. Una aplicación tan grande e importante como es CaixaBank necesita que exista un proceso de validación de datos de forma continua para la obtención de permisos con tal de que todo el ecosistema de desarrollo siga siendo seguro.

Sprint 3

Este tercer Sprint era el inicio del desarrollo de la principal operativa de negocio que se implementa en este trabajo de fin de grado. Dicha operativa es la de solicitar el anticipo de facturas de proveedores. Esta operación está formada por un flujo de cuatro pantallas.

Hasta ahora era un proceso complejo en el que el usuario debía estar muy atento a toda la información que aparecía en pantalla y había ciertos aspectos que podían resultar confusos, algo que un banco como CaixaBank no se puede permitir. Las aplicaciones de banca online siempre tienen que tener como principal objetivo reducir la complejidad y facilitar al usuario todo tipo de consultas y transacciones que desee realizar. Si cualquier aspecto resulta complejo o poco intuitivo, el usuario optará por no utilizar dicha aplicación y preferirá ir a la oficina o, bien, por analizar otras plataformas de terceras entidades bancarias que cumplan correctamente sus necesidades.

Es por todo esto que la nueva operativa de solicitar anticipos de las facturas de proveedores debe ser fácil de usar, muy intuitiva y a la vez que ofrezca todas las funcionalidades que sean necesarias al usuario.

Para poder definir unos requisitos óptimos para esta y para cualquier funcionalidad es imprescindible tener muy en cuenta quien será el usuario final del producto. En el caso de este proyecto será un cliente de CaixaBank que posea una empresa y que trabaje con proveedores. Por lo tanto, tiene que existir un proceso de análisis de este tipo de usuario tan específico. Ya que en este caso consiste en una mejora de una operativa existente y, por consiguiente, ya hay usuarios que han utilizado el producto, el primer paso del análisis de requisitos es detectar qué resulta incómodo actualmente a estos usuarios. La segunda fase, es detallar todo lo que se desea que tenga la aplicación en el futuro y como se van a solucionar los problemas localizados.

Por último, es vital que dichos requisitos se integren dentro del mismo plan de renovación del conjunto de operativas de CaixaBank y que todos los cambios sean consistentes entre sí [15]. Todas las distintas operativas de negocio tienen que tener ciertas características en común, pese a que sean totalmente diferentes en funcionalidad. Esto es algo básico e imprescindible para no confundir al usuario y que no tenga que aprender cómo funciona todo cada vez que acceda a una pantalla nueva por primera vez. Por el contrario, tiene que resultarle todo familiar y que sea capaz de dominar toda la aplicación de una forma sencilla y rápida.

CaixaBank | banca digital Caixa

CaixaBank Now

Estás en la versión demo. Hazte cliente de CaixaBank

(usuario demo) Hola Victor

Tesorería Tarjetas Inversiones Financiación Comercio exterior Servicios Ficheros Móvil Commerce

Comprar Estrella

Carta de efectos
Líneas de riesgo
Factoring
Confirming
Préstamos y créditos
Avalés
Leasing
Renting
Promociones inmobiliarias
Catálogo de productos

Confirming - Operativa proveedor - Solicitud de cobro anticipado

selección facturas selección cuenta abono confirmación resultado solicitud

Proveedor: VICTOR SAULER PORTAL
Cliente: Ayuntamiento Nueva Villa
NIF Proveedor: 123456782

Selecciona las facturas que quiere anticipar y pulse el botón Calcular

☐ Seleccionar todas las facturas mostradas

Factura	Fecha factura	Fecha vencimiento	Importe nominal	Tipo interés	Tipo comisión	Importe neto *
<input checked="" type="checkbox"/> 785/355	22/06/2009	01/11/2009	9.353,85 EUR	4,625%	Mínima	9.135,79 EUR
<input checked="" type="checkbox"/> 123/062	22/06/2009	01/11/2009	13.281,69 EUR	4,625%	Mínima	12.972,06 EUR
<input checked="" type="checkbox"/> 785/321	30/05/2009	01/10/2009	3.337,80 EUR	4,625%	Mínima	3.259,99 EUR
<input checked="" type="checkbox"/> 454/890	30/05/2009	01/10/2009	6.155,46 EUR	4,625%	Mínima	6.011,96 EUR
<input checked="" type="checkbox"/> 898/521	15/04/2009	15/09/2009	27.466,32 EUR	4,625%	Mínima	27.027,14 EUR
<input checked="" type="checkbox"/> 412/022	19/04/2009	01/09/2009	1.114,87 EUR	4,625%	Mínima	1.097,76 EUR
<input checked="" type="checkbox"/> 784/032	19/04/2009	01/09/2009	33.558,08 EUR	4,625%	Mínima	33.172,40 EUR
<input checked="" type="checkbox"/> 154/784	30/03/2009	29/08/2009	13.684,17 EUR	4,625%	Mínima	13.526,90 EUR
<input checked="" type="checkbox"/> 125/875	22/03/2009	01/08/2009	31.781,45 EUR	4,625%	Mínima	31.538,68 EUR
<input checked="" type="checkbox"/> 795/522	22/03/2009	01/08/2009	30.823,70 EUR	4,625%	Mínima	30.588,24 EUR

Inicio > Siguientes

* Importe neto a abonar en caso de solicitar el cobro anticipado a fecha de hoy

Solicitud momentánea. No se genera y se suspende de cuentas de préstamos y tarjetas de crédito.

Figura 7. Confirming Petición 19 - Versión antigua

CaixaBank | banca digital Caixa

CaixaBank Now

MANUEL - Cambiar usuario

MI gestor

DEBUG

Cuentas Cobros Pagos Tarjetas Financiación Import-Export Inversión Servicios Ficheros

Líneas de riesgo
Factoring
Confirming de proveedor
Confirming de cliente
Avalés
Préstamos y créditos
Financiación comercio exterior
Leasing
Renting

Solicitar anticipo | CONSTRUCTORA PARDO M

Datos Configurar Confirmar

	TOTAL ANTICIPAR	TOTAL FACTURAS	INTERESES	COMISIONES
	93,12 \$	100 \$	- 0,83 \$	- 5 \$

Selecciona las facturas que quieres anticipar

<input type="checkbox"/> Nº factura	Importe factura	Importe anticipo	Fecha vencimiento	Días hasta el vencimiento
<input type="checkbox"/> 54/2009	100,00 \$	94,17 \$	11/09/2020	97
<input checked="" type="checkbox"/> 54/2009	100,00 \$	94,17 \$	11/09/2020	97
<input type="checkbox"/> 54/2009	100,00 \$	94,18 \$	11/09/2020	97

← Volver Continuar > X Cancelar

Nuevo CaixaBank Now. Vuelve a ver el tutorial con todos los novedades aquí.

Figura 8. Confirming Petición 19 - Versión actual

En la siguiente figura se muestra la nueva pantalla. A simple vista, la nueva versión ofrece una visión de la información mucho más clara y precisa. Se separa cada componente de la pantalla con la intención de que el usuario pueda encontrar aquella información que busque de una manera mucho más rápida y sencilla.

Detengámonos ahora en analizar cada cambio con más detalle. En primer lugar, se simplifica el título de la pantalla. No se repite información de donde se encuentra el usuario, sino que el título se centra en informar cuál es la operativa actual, “Solicitar Anticipo”, y el nombre del cliente seleccionado anteriormente. Para encontrar esta misma información en la versión antigua de la pantalla habría que haber leído las tres partes del título y buscar el campo de “proveedor” en mitad de la página.

El siguiente componente de la vista que ha sido modificado es la barra de progreso. Se ha mantenido el componente en sí al ser un muy buen indicador para el usuario, ya que permite conocer en todo momento el estado en el que se encuentra dentro de la operativa. En la nueva versión se puede apreciar cómo se ha reducido la operativa en un paso. Se ha decidido extraer el resultado de la operativa de la barra de progreso ya que lo único que transmitía era una idea de que la operativa era de mayor complejidad.

El siguiente elemento de la antigua pantalla era una serie de campos de datos que se ha eliminado por completo. Los motivos han sido que el nombre del cliente ya se ha incorporado en el título como hemos comentado. Los otros dos datos eran el propio nombre y NIF del usuario que realizaba la operativa. Es un sin sentido que algo tan básico como estos campos se muestre en una pantalla y entorpezca el objetivo principal, que es seleccionar las facturas que se desean anticipar.

La tabla de facturas a anticipar se ha modificado por completo. Como hemos dicho, en esta pantalla se seleccionan todas aquellas facturas de las cuales se va a solicitar un anticipo. Es, por tanto, en esto donde la vista se debe centrar. Es necesario que el usuario sepa interpretar correctamente cualquier información que se le presente por lo que se ha reducido el número de columnas manteniendo a la vez el máximo de información. Las columnas que se han eliminado han sido la de tipo de comisión, tipo de interés y fecha factura. Para comodidad del usuario se ha añadido una columna donde se muestran el número de días hasta el vencimiento de la factura, de esta manera no tiene que calcular el cliente dicho valor.

Sprint 4

En el cuarto Sprint se continuó con la misma pantalla del tercer Sprint, la petición 19 del proyecto Confirming. Durante la anterior iteración se modificó el diseño de la vista antigua. Se realizaron diversos cambios, pero todos ellos fueron a nivel estético. El objetivo de esta iteración era dotar a la petición 19 de nueva funcionalidad.

Mejorar la experiencia de usuario no es tan solo mejorar estéticamente la web y que quién utilice la aplicación encuentre de manera sencilla aquello que antes podía resultar complejo o poco intuitivo. Sino que también la información que se ofrezca al usuario sea útil y de ayuda. Tiene que ir de la mano la mejora visual con la mejora de información o de funcionalidad.

Por estos motivos, se ha añadido un nuevo componente a la petición de seleccionar facturas. Éste es una calculadora en el que se muestra el valor total de los importes a anticipar, de los importes de las facturas, de los intereses y de las comisiones de todas y cada una de las facturas de la tabla que tenga el usuario seleccionadas.

Esta calculadora permite que el usuario con un simple clic sea capaz de saber cuál será el total del importe que recibirá en el caso de completar la solicitud y cuáles serán las comisiones e intereses que tendrá que abonar. Esto es una gran mejora para los usuarios de Confirming de CaixaBank que ahora podrán disponer de una nueva herramienta de ayuda para calcular, analizar y solicitar sus anticipos de facturas.

La calculadora se ha desarrollado mediante un script en Javascript que recoge la información de las facturas de la tabla. En el momento en el que se hace clic en una fila de la tabla se calculan los nuevos valores y se muestran en el componente html de la calculadora.

La otra fase del desarrollo de este Sprint se ha dedicado a implementar la paginación de la tabla de facturas. Este componente ha sido desarrollado en AJAX. Esta acción antes resultaba tediosa para el usuario debido a que se tenía que asegurar de qué facturas había marcado en todas las páginas de la tabla. Por lo tanto, se ha agrandado el máximo número de facturas que se visualiza en cada página de la tabla, concretamente de 10 a 25 facturas. Con esta decisión se ha querido que la acción de paginar se realice las mínimas veces posibles. Aun así, con la incorporación de la nueva calculadora resulta mucho más

sencillo llevar un control de aquellas facturas que se han seleccionado, sea en la página que sea.

Sprint 5

En el quinto Sprint se implementó la siguiente pantalla del flujo de la operativa de solicitud de anticipo de una factura que corresponde con la petición 21 del proyecto de negocio Confirming.

A esta petición se accede cuando se presiona el botón continuar de la petición 19, una vez se han seleccionado las facturas a anticipar. Consiste en seleccionar la cuenta bancaria en la que se realizará el ingreso del importe de las facturas anticipadas.

En las figuras 9 y 10 se muestra como era la versión antigua de la vista de la petición y en la figura 11 se muestra como es la versión actual. En primer lugar, el título se ha cambiado y presenta el mismo formato y contenido que en la nueva versión de la petición 19: operativa y nombre del cliente. Así mismo, la barra de progreso indica que esta pantalla corresponde a la segunda fase del flujo de negocio.

La anterior versión centraba prácticamente la totalidad de la vista en exponer un resumen de las facturas seleccionadas y del valor de los importes a anticipar, de facturas, de comisiones y de intereses entre otros. Incluso se dedicaba gran parte del *frame* a repetir la información del nombre y NIF del usuario y del nombre del cliente que ya aparecía en la pantalla anterior.

Esto provocaba que la selección de la cuenta bancaria, que era en principio el principal objetivo de la petición quedaba relegada a la parte más inferior de la vista. Teniendo incluso que el usuario realizar *scroll* o desplazamiento hacia abajo para poder detectar la selección de la cuenta.

Esto es algo que daña completamente la experiencia del usuario en la aplicación. Se bombardea de información, mucha de ella repetida, y relega la principal funcionalidad a lo último. Esto puede provocar confusión y complejidad a los usuarios y es algo que se ha tenido muy en cuenta a la hora de definir los requisitos para la nueva versión de la pantalla.

Por el contrario, la renovación de la pantalla cambia todo esto por completo. Como vemos en la figura 11, ahora se presenta una lista con las cuentas bancarias del usuario de la aplicación. En esta lista se muestra el código IBAN y el saldo actual que hay disponible en cada una de las cuentas. Se añade también otro listado igual que el anterior, pero con cuentas bancarias externas a CaixaBank y vinculadas al usuario. Se puede acceder haciendo clic en la opción de “Cuentas externas vinculadas”.

En la anterior versión solo era posible seleccionar una de las cuentas de CaixaBank mediante un elemento de selección. Sin embargo, ahora se ha añadido la comentada nueva funcionalidad de cuentas externas vinculadas y la opción de teclear cualquier número de cuenta que se desee. Estos cambios suponen un gran beneficio al usuario final ya que, hasta ahora, si se deseaba ingresar los anticipos en otras cuentas se debía tramitar la gestión desde las oficinas.

Confirming - Operativa del proveedor - Pedir cobro anticipado

Proveedor: VICTOR SAULER PORTAL, NIF: 123456782
 Cliente: Ayuntamiento Nueva Villa

Datos del cálculo del cobro anticipado de las facturas

Estado de la solicitud	Pendiente de confirmar
Importe nominal	252.906,47 EUR
Fecha de liquidación	27/07/2009
Intereses	2.223,27 EUR
sobre intereses	0,00 EUR
Tipo de interés (TAE)	4,625%
Comisiones	632,27 EUR
Número de facturas incluidas	10
sobre comisiones	101,16 EUR
Importe neto a abonar	249.949,77 EUR
Contravalor importe neto a abonar	135,50 EUR *

Seleccione la cuenta de abono

* Cuenta en el extranjero

Figura 9. Confirming Petición 21 - Versión antigua 1

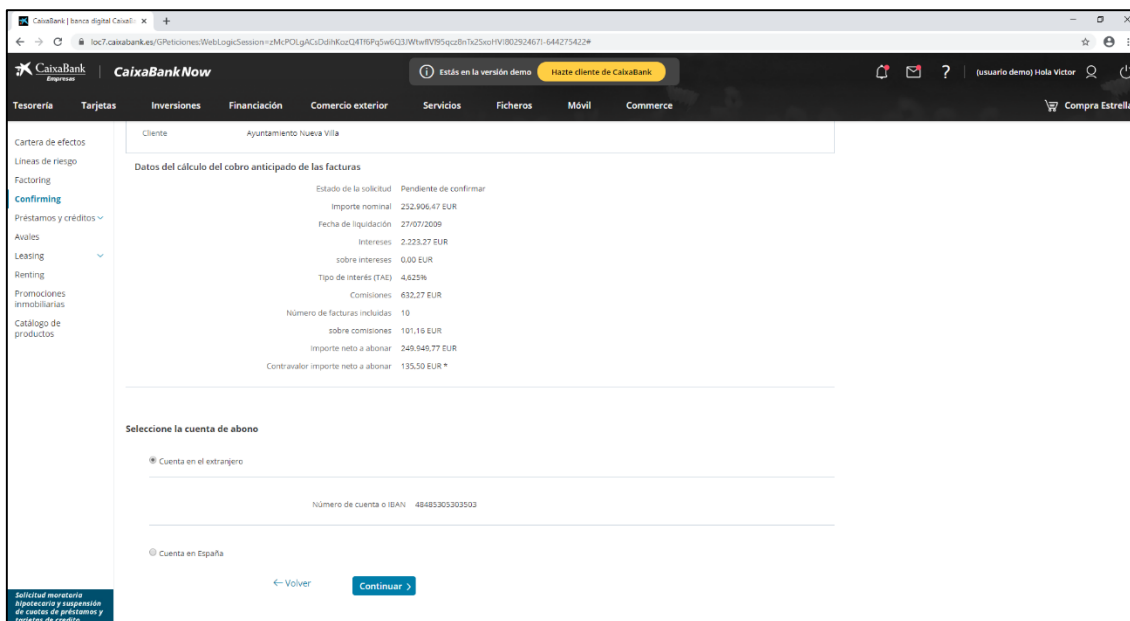


Figura 10. Confirming Petición 21 - Versión antigua 2

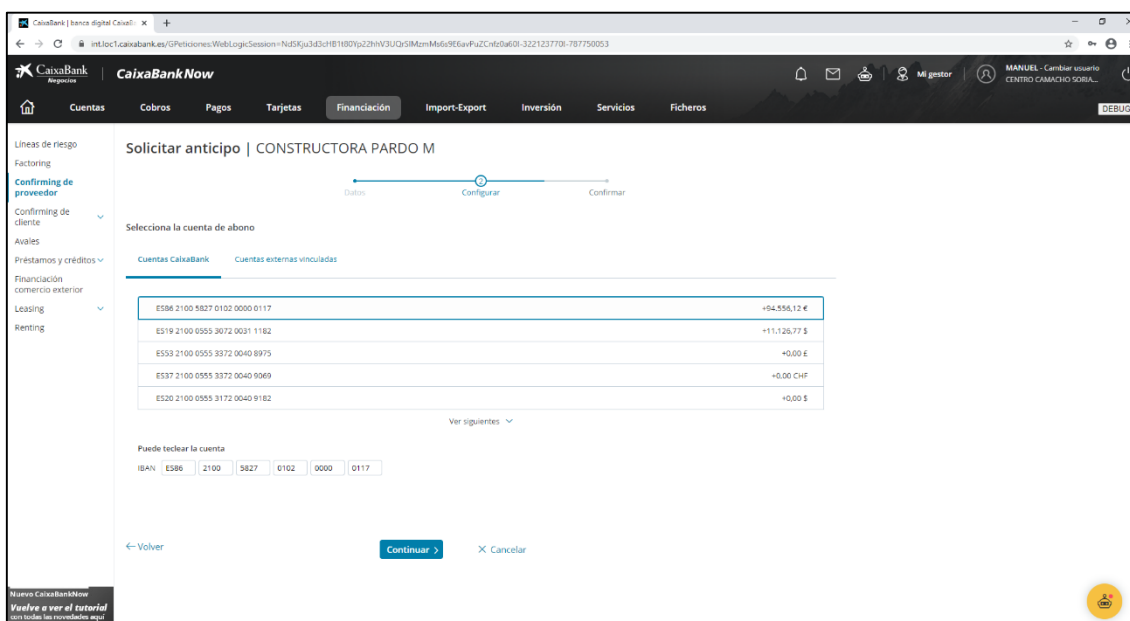


Figura 11. Confirming Petición 21 - Versión actual

Sprint 6

El sexto Sprint se dedicó a la implementación de la siguiente pantalla de la operativa de solicitar anticipo de una factura. Esta pantalla es la petición número 25 del proyecto de negocio Confirming.

Esta petición es la última fase de la solicitud y consiste en la confirmación de la misma. El objetivo de esta vista es mostrar al usuario un resumen de la operación que está en trámite y pedir un pin secreto con tal de completar y autorizar dicha solicitud.

La versión anterior de la petición se muestra en la figura 12. En ella se puede ver como la información de la operación se presenta a los usuarios en forma de lista de campos, una fila por cada uno de los datos. Esta disposición de la información provoca que no sea posible mostrarla y visualizar de forma completa en la mayoría de monitores.

Operativa proveedor - Solicitud de cobro anticipado - Confirmación	
Petición de cobro anticipado aún no finalizada. Compruebe que los datos son correctos y confirme la operación al final de la página.	
Datos generales	
Proveedor:	VICTOR SAULER PORTAL
Cliente:	Ayuntamiento Nueva Villa
Datos de la operación	
Tipo:	Solicitud de cobro anticipado
Fecha de liquidación:	27/07/2009
Tipo de interés (TAE):	4,62%
Número de facturas:	10
Número de cuenta o IBAN:	48485305303503
Importe nominal:	252.906,47 EUR
Intereses:	2.223,27 EUR
Comisiones:	692,27 EUR
I.V.A. sobre comisiones (16%):	101,16 EUR

Confirmar la operación: Busque el número 230 y pulse la clave correspondiente en el teclado siguiente:

Núm. Clave: 230

Confirmar operación Cancelar

Figura 12. Confirming Petición 25 - Versión antigua

Por lo que el usuario debe realizar un *scroll* o desplazamiento hacia abajo con el ratón para poder apreciar todos los datos. Como hemos comentado anteriormente, es mejor que toda la información importante se muestre en el centro de la pantalla y que todos los usuarios puedan ser capaces de localizarla y entenderla con facilidad.

Analizando estos problemas que presenta la que hasta ahora era la última versión de la petición procedemos a realizar una modificación para mejorar la experiencia del usuario.

Esta es una petición de confirmación de una solicitud bancaria muy importante y crítica. Por esto es vital que el usuario entienda con facilidad que es lo que está a punto de autorizar.

Se ha decidido presentar los datos en formato de columnas en el *frame*. De esta manera se dispone de toda la información en el centro de la pantalla y resulta muy sencillo ver todos los valores a simple vista. Otra decisión importante ha sido la de seleccionar únicamente aquellos datos realmente importantes de todos los que se estaban mostrando en la versión anterior.

CaixaBank | banca digital Caixa | x

int:loc1:caixabank.es/GPeticionesWeb?logicSession=Nd5Kj3d3d1B1807(p22HvY3UQdSiMconMuf69E6wFuZCnfoa60i-322123770i-787750053)

CaixaBank Now

Cuentas Cobros Pagos Tarjetas **Financiación** Import-Export Inversión Servicios Ficheros

MANUEL - Cambiar usuario CENTRO COMERCIO SORAL...

DEBUI

Lineas de riesgo

Factoring

Confirming de proveedor

Confirming de cliente

Avales

Préstamos y créditos

Financiación

comercio exterior

Leasing

Renting

La solicitud de anticipo aún no ha finalizado. Comprueba los datos y confirma la operación

Datos generales

Cliente	Cuenta de abono
CONSTRUCTORA PARDO M	ES19 2100 0555 3072 0031 1182

Datos del anticipo

Total anticipo	Comisiones	I.V.A sobre comisiones (21%)
92,12 \$	5,00 \$	1,05 \$
Número de facturas	Tipo de interés (25,93%)	I.V.A sobre intereses
1	0,83 \$	0,00 \$
Total facturas		
100,00 \$		

Introduzca su número secreto de autorización

Número secreto de autorización ¿Qué es?

Confirmar operación X Cancelar

Nuevo CaixaBankNow

Vuelve a ver el tutorial con todos los novedades aquí

Figura 13 . Confirming Petición 25 - Versión actual

Sprint 7

Una vez finalizada las nuevas pantallas de las tres fases de la operativa de solicitud del anticipo de una factura se pasó a implementar la vista de resultado de la operación. Esta es la petición 27 del proyecto Confirming.

Esta petición consiste en notificar a los usuarios de la aplicación si la solicitud del cobro anticipado de sus facturas se realizó con éxito o, por el contrario, sucedió algún problema. Si comparamos la figura 14, que muestra la versión antigua, y la figura 15, que evidencia la nueva versión, distinguimos lo siguiente.

En cuanto a diseño y apariencia cambia la forma en la que se presentan los datos de la solicitud. Se ha modificado las filas de datos por un recuadro extensible que CaixaBankNow está utilizando recientemente también en el resultado de otras operativas. Sin embargo, la parte superior de la interfaz se ha decidido mantener prácticamente igual. El *check* verde es una gran forma de reflejar a los usuarios el éxito de la petición.

La funcionalidad, no obstante, sí que ha cambiado en gran medida. Ahora existen dos nuevos botones en la parte inferior del recuadro del resumen del anticipo. El botón de la derecha devuelve el control de la aplicación a la lista de clientes que disponga el usuario para que sea capaz de realizar otro anticipo fácilmente. Esta funcionalidad es una redirección muy simple a otra vista. Sin embargo, el botón de la izquierda presenta la incorporación de una funcionalidad totalmente nueva para los clientes Empresa de CaixaBankNow.

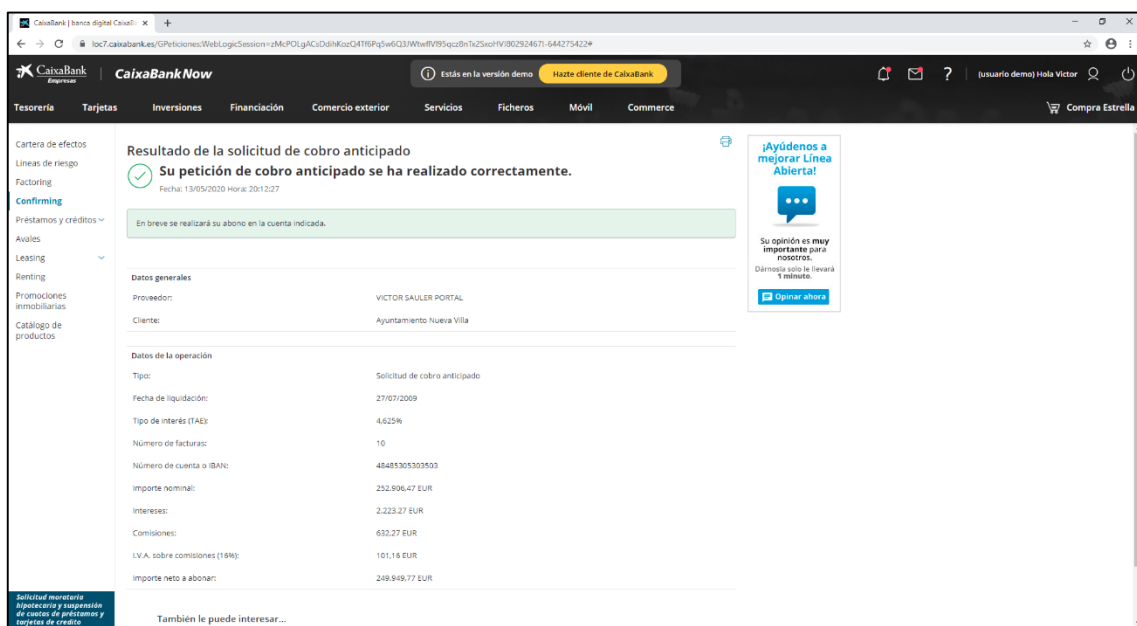


Figura 14. Confirming Petición 27 - Versión antigua

Esta funcionalidad es la de poder automatizar los próximos anticipos de facturas del cliente. Esto abre la oportunidad a los usuarios de agilizar sobremanera todo este proceso y, simplemente, indicar a la aplicación cuando y como se desea solicitar un anticipo de facturas. Esto es algo que todavía no está implementado, pero sí que se ha tenido en cuenta en el desarrollo de este proyecto. Por tanto, gracias a estar definido el requisito, hemos

dejado estructurado y maquetado todo lo necesario en esta petición para que, cuando esté creada la vista de automatizar anticipo, se redirija desde aquí enviando toda la información necesaria.

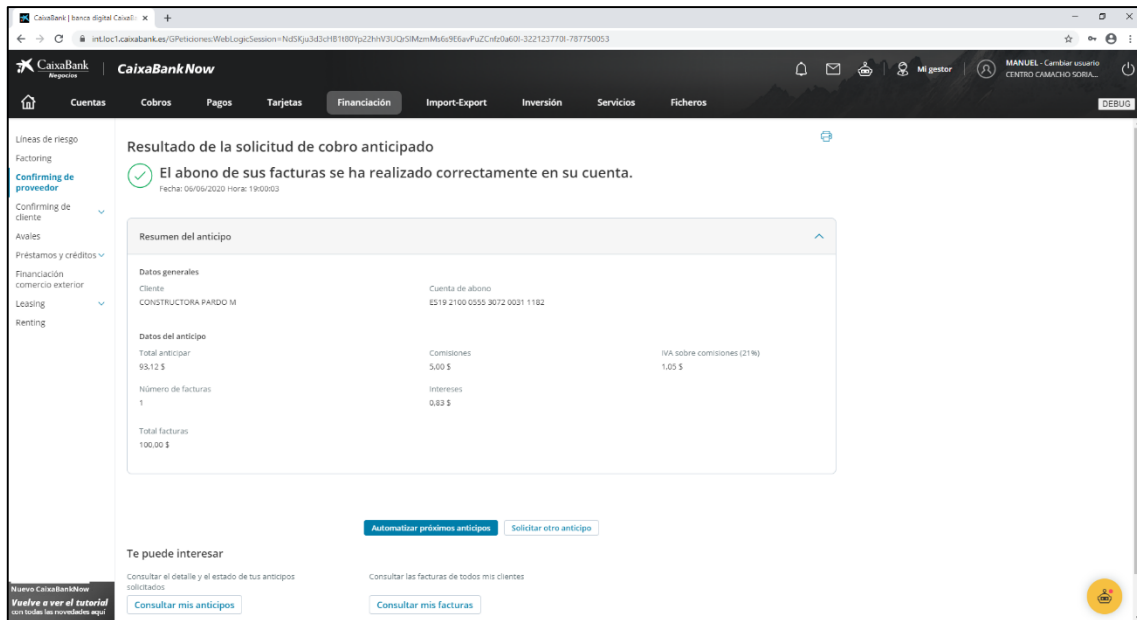


Figura 15. Confirming Petición 27 - Versión actual

Sprint 8

Esta iteración del desarrollo del proyecto fue distinta a las anteriores. Como he explicado, los anteriores Sprints se dedicaban a la implementación de una petición. En cambio, el principal objetivo de estas semanas fue el testing de todos los servicios de backend del proyecto de negocio Confirming.

Junto con la renovación del frontend del Confirming de CaixaBankNow, que es en lo que consiste este trabajo de fin de grado, se han creado nuevos servicios en backend para poder ofrecer toda la información que se muestra en la web. Este desarrollo de backend ha sido llevado a cabo por otro equipo de trabajo. En este Sprint se ha realizado el testing y la validación de la integración de las vistas con los nuevos servicios.

Durante las anteriores iteraciones había disponibles algunos de los servicios, aunque no la totalidad de ellos. Para poder realizar el desarrollo de frontend sin la necesidad de esperar a que esos servicios estuvieran disponibles, se disponía de un usuario “Demo” que tiene información ya guardada y no realiza llamadas a backend para obtener los datos.

Siempre es importante que en cualquier proyecto se realicen actividades de Quality Assurance para poder validar los requisitos que se han planteado inicialmente. A pesar de que existe un equipo de trabajo dedicado especialmente a esta práctica, es interesante que también participen los propios desarrolladores en el testing y validación de requisitos y de los criterios de aceptación.

Las pruebas tuvieron un resultado positivo en la gran mayoría de casos. Se detectaron ciertos errores en las llamadas para obtener los intereses y comisiones de la petición 19. Esto surgió porque ha cambiado la forma de calcular esos importes y esos cambios no se veían reflejados en la web. Una vez corregido esos defectos se repitieron las pruebas y se dieron por superadas al finalizar todas con éxito.

Sprint 9

El último Sprint del proyecto se dedicó para tareas de análisis web. Una vez terminada toda la implementación del proyecto se procedió a incluir etiquetas (tags) en el código. Estas etiquetas son pequeños fragmentos de código Javascript que recolectan información de las acciones que realiza el usuario en la web para su posterior análisis.

Los análisis permiten entender el público que accede a la web, así como medir los resultados de marketing digital. Es posible realizar un seguimiento de varias informaciones sobre tu dominio: como el volumen de visitas, el tiempo que el usuario emplea en las páginas, donde acceden los usuarios con más frecuencia o el tiempo entre las diferentes acciones.

La recolección y el análisis de la información es, obviamente, responsabilidad de CaixaBank. Aun así, para poder facilitar la gestión de las etiquetas y su mantenimiento se ha decidido añadir un tag manager que permite unificar las diferentes herramientas en una sola y disponer de una sola interfaz de trabajo. Este tag manager se llama Tealium. Por lo tanto, en este Sprint principalmente se han añadido diferentes etiquetas en todas las nuevas vistas desarrolladas en el nuevo proyecto y en las versiones anteriores.

11.2. Testing

El testing resulta una fase muy importante del desarrollo de una historia de usuario. Nos permite validar el código escrito, poder identificar errores en la aplicación y valorar la capacidad de corregirlos. A parte de trabajar pruebas exhaustivas de los servicios durante el noveno Sprint, en cada iteración ha existido un desarrollo de test y de corrección del código escrito.

Para poder comprobar el correcto funcionamiento de la lógica de negocio en las operativas de negocio se ha decidido utilizar el framework JUnit. Con esta herramienta podemos crear test unitarios de cada una de las clases java implementadas. Además, se ha utilizado el framework Mockito para facilitar la creación de objetos que simulan el comportamiento de objetos reales. Esto resulta muy útil para el desarrollo de test unitarios.

Para poder desplegar un paquete en el entorno de integración de la Caixa es necesario que se ejecuten con éxito todos los tests realizados y que éstos tengan una mínima cobertura del 90% del código implementado. De no cumplirse alguna de estas restricciones, la herramienta de despliegue continuo lanzará un error y rechazará el despliegue del paquete en cuestión.

Por otro lado, también se utiliza en el proyecto la herramienta previamente descrita llamada SonarQube. Esta plataforma analiza la calidad del código de cada proyecto java de la aplicación web. De esta manera, se puede llevar a cabo un control exhaustivo de todos los proyectos e ir mejorando continuamente los errores implementados. De la misma forma que sucedía con las pruebas unitarias, se ha establecido ciertos límites de errores detectados por SonarQube para que se despliegue un paquete en el entorno de desarrollo. Concretamente, solo podrá haber como máximo diez problemas (*issues*) de gravedad crítica y cero de gravedad bloqueante en todo un proyecto de negocio. Pese a que estas sean las restricciones que se imponen para poder integrar el código implementado, se ha trabajado y corregido *issues* durante todo el proyecto para que los fallos en el análisis del código sean los mínimos posibles.

11.3. Integración y entrega continua

La integración continua es una práctica del desarrollo de software en la que los miembros de un equipo integran su trabajo a diario. Esta praxis permite reducir los problemas de integración y hace que el desarrollo sea más ágil y rápido.

La entrega continua hace referencia a una práctica de desarrollo de software que amplía el concepto de la integración continua. Consiste en preparar automáticamente el código y desplegar todos los cambios realizados en un entorno de pruebas o de producción después de la fase de compilación. De esta forma, la entrega continua permite disponer en todo momento de un artefacto listo para su despliegue el cual se ha sometido a un proceso de pruebas estandarizado.

En este proyecto se ha querido, desde el primer momento, utilizar estas prácticas de desarrollo. Para poder efectuar estas metodologías de trabajo se ha utilizado la herramienta llamada Jenkins.

Jenkins ofrece una manera simple de configurar un entorno de integración continua y de entrega continua. Se puede construir e integrar de forma rápida y robusta toda la cadena de herramientas de compilación, test e implementación.

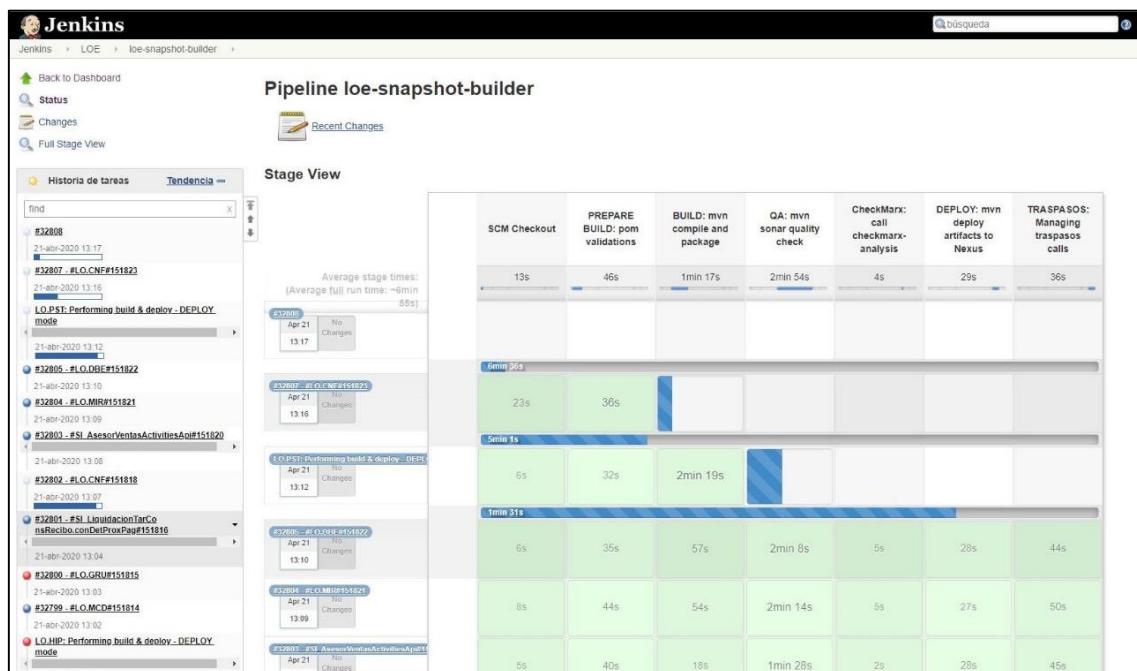


Figura 16. Jenkins

La metodología seguida por todo el equipo de proyecto ha sido la misma. Integrar a diario lo máximo posible los cambios realizados por cada uno de los miembros. Para esto, se subían el nuevo código implementado y las modificaciones realizadas a un repositorio de la herramienta de control de versiones Subversion. A partir de ahí, Jenkins era el encargado de validar, compilar, construir un nuevo paquete y de realizar todas las pruebas del código y de entregar a SonarQube el nuevo código para su correspondiente análisis.

En cuanto a la entrega continua, ésta estaba limitada a tres momentos a lo largo del día. Es decir, en ciertas horas se producía una entrega al entorno de Integración. Estas horas eran a las nueve de la mañana, a las dos y a las cinco de la tarde. Los ejecutables que se despliegan en dicha entrega son todos aquellos que, en el momento de la subida al repositorio, se han marcado mediante un atributo en el fichero .pom para que así sea.

12. Conclusiones

Llegados al final del proyecto, es momento de preguntarse si se han conseguido alcanzar los objetivos planteados, consolidar las competencias técnicas definidas al inicio del trabajo y comentar cuál sería el trabajo futuro que se podría realizar para continuar el proyecto.

Por lo que hace referencia a los objetivos del proyecto, creo que se han cumplido con éxito todos los objetivos inicialmente establecidos. Se ha logrado renovar totalmente una operativa de negocio de una entidad financiera muy importante como lo es CaixaBank y, además, mejorando la experiencia de los usuarios en la aplicación.

En este proyecto he tenido la oportunidad de aplicar muchos conceptos claves de la especialidad de Software como son los patrones de diseño, modularización del código, el análisis de requisitos y la gestión de grandes proyectos. Desde un punto de vista personal me ha resultado muy reconfortante conseguir integrar, poner en práctica y consolidar conceptos y conocimientos aprendidos durante la carrera. La posibilidad de haber podido aplicarlos en un proyecto para un entorno real de trabajo me ha permitido ganar confianza, seguridad, experiencia y capacidad para afrontar nuevos proyectos laborales en el futuro.

Poder realizar este proyecto en convenio con la empresa Indra, me ha permitido conocer cómo se trabaja en una gran empresa y dentro de un gran equipo de trabajo. He tenido la oportunidad de trabajar con metodologías ágiles, interaccionar con los miembros del equipo humano de trabajo, participar en reuniones con el cliente y colaborar en las planificaciones de los distintos Sprints del proyecto.

La propia elaboración y redacción de este documento me ha permitido tener una visión global del proyecto, reconocer la relevancia del posible impacto medioambiental y la importancia de gestionar y planificar temporalmente el trabajo. También he podido ejercitar la planificación económica y evaluar la viabilidad del proyecto software.

Destacar que este trabajo de colaboración se inició de forma presencial en Indra y se finalizó de forma telemática debido a las excepcionales condiciones impuestas por la pandemia por Covid-19. A pesar de las dificultades he sido capaz de resolver y ejecutar de forma satisfactoria las tareas encomendadas por los responsables del proyecto de CaixaBank.

Para finalizar, destacar que me ha parecido un proyecto muy interesante por el tema que desarrollaba. Pero, sobre todo, porque he podido aprender nuevas técnicas de programación, conocer la importancia de la planificación del tiempo, desarrollar una comunicación efectiva con los miembros del equipo de trabajo tanto en condiciones de presencialidad como de teletrabajo.

12.1. Consolidación de las competencias técnicas

- **CES1.1:** *Desarrollar, mantener y evaluar sistemas y servicios software complejos y/o críticos.*

He podido trabajar completamente esta competencia debido a haber participado en el desarrollo de un proyecto de una aplicación bancaria. En este tipo de empresas se trabaja con información personal y confidencial, donde la seguridad es un valor decisivo y determinante. El software que he analizado y reconstruido durante este proyecto es complejo y atiende operaciones y transacciones bancarias críticas. He podido evaluar el código con la herramienta SonarQube y detectar errores y posibles vulnerabilidades que puedan afectar al sistema.

- **CES1.3:** *Identificar, evaluar y gestionar los riesgos potenciales asociados a la construcción de software que se puedan presentar.*

Este proyecto es una renovación de un software construido hace años y que presentaba carencias. He estado trabajando en reconstruir parte de este software, ampliándolo y actualizándolo. Por consiguiente, me ha servido para poder identificar y evaluar con profundidad los riesgos que se pueden presentar en la construcción de una aplicación. Este proceso de análisis me ha servido a la hora de la implementación y me servirá para futuros proyectos. Siempre se tiene que tener en cuenta qué es lo que se está desarrollando, quién va a mantener el código y tratar, en lo posible, de minimizar y corregir todos los problemas que puedan surgir. No comentar el código, realizar apañes para

resolver problemas o no estructurar el software son ejemplos que pueden provocar serios problemas en la mantenibilidad de éste.

- **CES1.7:** *Controlar la calidad y diseñar pruebas en la producción de software.*

Durante toda la fase de implementación se ha trabajado esta competencia con la creación de tests unitarios con JUnit y Mockito. Pero, sobre todo, se ha practicado en profundidad y rigor durante el Sprint 9 en la fase de testing y validación de servicios. Es algo básico en cualquier proyecto de software y he podido dedicar una iteración completa para consolidar esta competencia técnica.

- **CES2.1:** *Definir y gestionar los requisitos de un sistema software.*

Para el desarrollo de toda la operativa de anticipo de facturas ha existido una fase de definición y discusión de requisitos para cada petición con cliente. Además, también se han ido corrigiendo algunos requisitos definidos conforme iba avanzando la implementación. La gestión de los requisitos se ha podido trabajar también con herramientas de seguimiento como lo son Jira y Confluence.

- **CES1.9:** *Demostrar comprensión en la gestión y gobierno de los sistemas software.*

Gracias a participar en un proyecto para una entidad financiera tan importante como CaixaBank y en una aplicación tan significativa como CaixaBankNow he podido comprender como se gestiona, se controla y se organiza un sistema software de grandes dimensiones. Es importante poder dedicar diferentes equipos de desarrollo para las distintas secciones de producción. Sin embargo, el aspecto más importante que he podido apreciar es el tener una buena comunicación entre todos los equipos y una buena capacidad de organización y desarrollo.

12.2. Trabajo futuro

En este proyecto se ha renovado por completo la operativa de solicitar el anticipo de una factura de la plataforma bancaria CaixaBankNow. Pese a que se ha completado el desarrollo total de esta operativa, existe gran cantidad de trabajo para realizar en el futuro.

En primer lugar, como hemos visto en el Sprint 7 de este trabajo, se quiere incluir en la aplicación la funcionalidad de automatizar la solicitud de anticipos. Esto es algo que aún no está completamente desarrollado y que es necesario crear nuevas peticiones en el proyecto de Confirming para poder resolver e integrar este nuevo requisito funcional. Por lo tanto, este sería un gran trabajo con el que poder continuar y extender el proyecto.

Otra operativa relacionada con este proyecto es la de anular un anticipo. CaixaBank quiere también renovar todas las peticiones relacionadas con la anulación de anticipos por lo que es otro apartado con el que seguir mejorando y actualizando la aplicación.

13. Referencias

[1] “Facultat d’Informàtica de Barcelona” [en línea]. [Consulta: 20 febrero 2020]. Disponible en: <<https://www.fib.upc.edu/>>

[2] “Minsait” [en línea]. [Consulta: 20 febrero 2020]. Disponible en: <<https://www.minsait.com/es>>

[3] “CaixaBankNow | Banca digital | CaixaBank” [en línea]. [Consulta: 20 febrero 2020]. Disponible en: <https://www.caixabank.es/particular/bancadistancia/caixabanknow_es.html>

[4] “Confirming | Financiación | Empresas” [en línea]. [Consulta: 23 marzo 2020]. Disponible en: <https://www.caixabank.es/empresa/financiacion/caixaconfirming_es.html>

[5] “Why UX Design Is A Strategic Business Tool - Design Leadership Notebook - Medium” [en línea]. [Consulta: 13 junio 2020]. Disponible en: <<https://medium.com/design-leadership-notebook/why-ux-design-is-a-strategic-business-tool-9184c0feb6c6>>

[6] “Silk – Portal de la Filial Tecnológica de CaixaBank” [en línea]. [Consulta: 21 febrero 2020]. Disponible en: <<https://www.silk.es/>>

[7] “Jira | Software de seguimiento de proyectos e incidencias” [en línea]. [Consulta: 22 febrero 2020]. Disponible en: <<https://www.atlassian.com/es/software/jira>>

[8] “Confluencia: conseguir más juntos | Atlassian” [en línea]. [Consulta: 22 febrero 2020]. Disponible en: <<https://www.atlassian.com/es/software/confluence>>

[9] “Pràctiques en empresa | Facultat d’Informàtica de Barcelona” [en línea]. [Consulta: 28 febrero 2020]. Disponible en: <<https://www.fib.upc.edu/ca/empresa/practiques-en-empresa>>

[10] “Online Gantt Chart Software | TeamGantt” [en línea]. [Consulta: 14 marzo 2020]. Disponible en: <<https://www.teamgantt.com/>>

[11] “Sueldos en Indra | Glassdoor.es” [en línea]. [Consulta: 7 marzo 2020]. Disponible en: <<https://www.glassdoor.es/Sueldo/Indra-Sueldos-E9757.htm>>

[12] “Spaces 22 Arroba | Centro de negocios con coworking en Barcelona” [en línea]. [Consulta: 7 marzo 2020]. Disponible en:

<<https://coworkingspain.es/espacios/coworking/barcelona/spaces-22-arroba>>

[13] “Code quality and Security | SonarQube” [en línea]. [Consulta: 22 febrero 2020]. Disponible en: <<https://www.sonarqube.org/>>

[14] “What Is the ROI of UX Design? | Clutch.co” [en línea]. [Consulta: 14 junio 2020].

Disponible en: <<https://clutch.co/app-developers/resources/what-roi-ux-design#:~:text=For%20every%20dollar%20invested%20in,and%20leave%20room%20for%20growth.>>

[15] “Design principle: Consistency - UX Collective” [en línea]. [Consulta 16 junio 2020]. Disponible en :<https://uxdesign.cc/design-principle-consistency-6b0cf7e7339f>

Anexo A. Diagrama de Gantt

